SSL Sertifikaları ile NTP'nin Güvenliğine Yeni Bir Yaklaşım

Bu tez Bilgi Güvenliği Mühendisliği'nde
Tezli Yüksek Lisans Programının bir koşulu olarak

Mustafa KÖĞÇE

tarafından

Fen Bilimleri Enstitüsü'ne
sunulmuştur.

Bu tezi okuduk, kapsam ve nitelik açısından Bilgi Güvenliği Mühendisliği alanında Yüksek Lisans derecesi için tümüyle uygun olduğu görüşüne vardık.

ONAYLAYANLAR:

Prof. Dr. Ensar Gül .....
(Tez Danışmanı)

Prof. Dr. Nizamettin Aydın .....

Dr. Öğretim Üyesi İhsan Çiçek .....

Bu tez İstanbul Şehir Üniversitesi, Fen Bilimleri Enstitüsü tarafından belirlenen tüm koşullara uygundur.

ONAY TARİHİ: 26.08.2019

MÜHÜR/İMZA:

Yazarlık Beyanı

Ben, Mustafa KÖĞÇE, başlığı, 'SSL Sertifikaları ile NTP'nin Güvenliğine Yeni Bir Yaklaşım' olan tezin ve içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Esaslı yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

İmza: 26.08.2019

Tarih:

SSL ile NTP'nin Güvenliğine Yeni Bir Yaklaşım

Mustafa KÖĞÇE

Öz

Zaman ve Zaman senkronizasyonu, özellikle zamana duyarlı işlemleri yapan bilgisayar ağları için önemlidir. Tüm veri merkezleri, borsalar, finans kuruluşları, endüstriyel ağlar, ticari uygulamalar, e-posta ve iletişim ile ilgili istemci ve sunucular, aktif dizin hizmetleri, kimlik doğrulama mekanizmaları, kablolu ve kablosuz iletişim için son derece önemlidir.

NTP, bir ağdaki tüm cihazları senkronize etmek için tek bir zaman kaynağı kullanarak çalışır. Bilgisayar ağları birbirleriyle zaman dilimleri arasında ve sık sık dünyanın diğer tarafından iletişim kurarken, bu ana zamanın sadece her yerde olması gerekmemekte, aynı zamanda oldukça hassas olması gerekmekte, aksi halde dünyanın farklı yerlerinde ağlar farklı zamanlarda çalışabilir.

NTP gibi yaygın internet protokolleri protokolün işleyişine zarar vermek ya da hizmeti kesintiye uğratmak için rutin olarak saldırgan girişimlere maruz kalabilmektedir. Bu çalışmada protokolün aksatabilecek wiretap (araya girme) ve middleman (aracı) işlemlerinden korunmak adına simetrik anahtar ve açık anahtar kriptografisi de dâhil olmak üzere NTP'nin güvenliğinin geliştirilmesi ve zaman bilgisinin kopyalanarak, değiştirilerek ve tekrarlamak suretiyle yapılacak ataklara karşı koruma sağlanması amaçlanmıştır.

Anahtar Sözcükler: NTP, DKIM, SSL, MTIM, RSA, ECDSA, NTP Atakları, Siber Güvenlik, Bilgi Güvenliği, Network, UDP

SSL Sertifikaları ile NTP'nin Güvenliğine Yeni Bir Yaklaşım

Mustafa KÖĞÇE

Abstract

The Time and the Time Synchronization are very important especially for the computer networks performing time-sensitive operations. It is very important for all the datacenters, exchange markets, finance companies, industrial networks, commercial applications, e-mail and communication-related clients and servers, active directory services, authentication mechanisms, and wired and wireless communication.

The NTP acts as a single time source in order to synchronize all the devices in a network. While the computer networks communicate with each other between different time zones and different locations on the earth; the main time doesn't need to be the same all around the world but it must be very sensitive otherwise the networks at different locations might work on different times.

Keywords: NTP, DKIM, SSL, MTIM, RSA, ECDSA, NTP Attacks, Cyber Security, Information Security, Network, UDP

Bu çalışma eşim Ayşe Namlı Köğçe ve oğlum Agâh Köğçe'ye ithaf edilmiştir.

Teşekkür

Tez çalışmam boyunca bana yol gösteren ve yardımlarını esirgemeyen saygıdeğer hocalarım Prof. Dr. Ensar Gül, Necati Ersen Şişeci ve benden hiçbir zaman desteğini eksik etmeyen eşim Ayşe Namlı Köğçe'ye ve aileme sonsuz teşekkürlerimi sunarım.

İçindekiler

Şekil Listesi

Tablo Listesi

Kısaltmalar

NTP	**N**etwork **T**ime **P**rotocol
UTC	**C**oordinated **U**niversal **T**ime
WAN	**W**ide **A**rea **N**etwork
LAN	**L**ocal **A**rea **N**etwork
GPS	**G**lobal **P**ositioning
ICRF	**I**nternational **C**elestial **R**eference **F**rame
ERA	**E**arth **R**otation **A**ngle
TAI	**I**nternational **A**tomic **T**ime
GMT	**G**reenwich **M**ean **T**ime
UDP	**U**ser **D**atagram **P**rotocol
MD5	**M**essage **D**igest **A**lgorithm 5
DNS	**D**omain **N**ames **S**ervice
BGP	**B**order **G**ateway **P**rotocol
RFC	**R**equest **F**or **C**omment
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
IP	**I**nternet **P**rotocol **A**ddress
MITM	**M**an in **T**he **M**iddle
SYN	**S**ynchronize
DDoS	**D**istrubuted **D**enial **O**f **S**ervices
RPKI	**R**esource **P**ublic **K**ey **I**nfrastructure
MAC	**M**essage **A**uthentication **C**ode
PBX	**P**rivate **B**ranch **E**xchange
ECDSA	**E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm
DSA	**D**igital **S**ignature **A**lgorithm
PoC	**P**roof **O**f **C**onsept

Fiziksel Katsayılar

Pi Sayısı $\Pi = 3.14$

Simgeler

Simge	İsim
μ	mikro
π	pi sayısı
δ	delta
θ	teta

Bölüm 1

Giriş

İnternet üzerinde faaliyet gösteren NTP hizmet protokolü, protokolün işleyişini ya da taşıdığı veriyi sekteye uğratmayı amaçlayacak pek çok saldırı türüne karşı savunmasız olabilmektedir. Bu çalışmada NTP güvenliğine dair bir analiz yapılmakta ve saldırganın yararlanmayı deneyebileceği bir dizi saldırı senaryosu dikkate alınmaktadır. Bu konu tüm serverlar ve istemciler için geçerli olmaktadır.

Her hangi bir kriptografik yöntem ile doğruluğu ispat edilebilen bir sunucu, eşitleme kaynakları ile ilgili olarak doğru zamanı ve mitigation algoritmalarına göre en iyi tahmini sunabilmektedir. Ancak doğruluğu kanıtlanmış bir sunucu senkronizasyon kaynaklarından gelen zaman bilgisi yanlış ise yanlış zaman verebilir. Öte yandan doğruluğu kanıtlanmamış olan bir sunucu ise doğru zamanı verebilir veya vermeyebilir hatta herhangi bir kaynak ile senkronize olmayan sahte bir zaman bile verebilir. Güvenirliği belirlemek için güvenilir tüm güvenlik tehditlerinin değerlendirilmesi gerekir. Bu çalışmada kullanıldığı şekliyle, belirli bir tehdite karşı savunma yapmak, bunun tespit edilmesi ve buna karşı önlemler açıklamaktadır.

Bu çalışmada ilk olarak ntp protokolü ve çalışma methodları ele alınarak ntp güvenlik modeli ayrıntılı olarak analiz edilmiştir.[18] Çalışmamızda bu güvenlik modelinin güvenliğinin arttırılmasına yönelik yeni bir yaklaşım sunmaktadır.

Bölüm 2

İlgili Çalışmalar

NTP güvenlik modeli, 4. bölümde şekil 4.1'de gösterilen hiyerarşik yapıya sahiptir. Saldırganın saldırısına karşı yürütülen savunma hiyerarşinin en alt tabakası olan On-Wire Protokolünde başlar. Bir üst tabaka olan Message Digest tabakası tarafından simetrik anahtar şifrelemesi kullanarak bu katmana yapılan saldırılara karşı savunma işlemi gerçekleştirilir. Paketleri sayısal imzalara bağlamak için bir hash veya pseudo-random sekans tekniği kullanan Autokey sequence tabakası ile savunma devam ettirilir.

Belirli bir tehdit bir katmanda başarıyla bertaraf edildiğinde, daha üst katmanlarda da savunulduğu varsayılmaktadır. Ancak, belirli bir tehdit bir katmanda başarıyla savunulmazsa, bu tehdit daha alt katmanlarda savunulabilir veya savunulmayabilir. Genel olarak, On-Wire Protokol katmanı üzerindeki katmanlar isteğe bağlıdır. Simetrik anahtar şifrelemesi sadece alt iki katmanı kullanır, açık anahtarlı şifreleme ise bütün katmanlar da kullanır. [18]

Standart konfigürasyonunda iken NTP paketlerinin korunmasız bir şekilde server ve istemci arasında alışverişi yapılabilir. Man-In-The-Middle olabilen saldırgan NTP paketinin içeriğini düşürebilir, yeniden uygulayabilir ya da değiştirebilir ki bu da zaman senkronizasyonunda bozulmaya ya da hatalı zaman bilgisinin iletimine neden olmaktadır. Zaman senkronizasyon protokolleri için ciddi bir tehlike analizi RCF 7384 tarafından sunulmaktadır. NTP paketlerinin özgünlük ve bütünlüğünü korumak için NTP iki iç güvenlik mekanizması sunmaktadır. Her ikisi de NTP paketini Mesaj özgünlük Kodu (MAC) ile korumaktadır. [39]

RFC 5905 için NTP'nin yayın modu simetrik anahtar kriptografisi kullanarak doğrulanmaktadır. Yayın sunucusu ve tüm yayın istemcileri simetrik bir kriptografik anahtar paylaşırlar ve yayın serverı bu anahtarı kullanarak mesaj doğrulama kodunu (MAC) gönderdiği yayın paketlerine iliştirir.

Burada önemli bir nokta da serverı dinleyen tüm yayın istemcilerinin bu kriptografik anahtarı biliyor olmaları gerektiğidir. Bu da herhangi bir istemcinin bu anahtarı kullanarak yayın serverından geliyormuş gibi görünen geçerli yayın mesajları iletebilecek olmalarıdır. Bundan ötürü de sahte yayın istemcisi diğer yayın istemcilerine saldırmak için bu anahtara dair bilgisini kullanabilir.

Bu nedenle NTP yayın serverı ve tüm istemcileri birbirlerine güvenmelidir. Yayın modu sadece güvenilir bir ağda çalıştırılmalıdır.

RFC 5905 ise MAC hesaplaması için desteklenmesi gereken bir hash tanımlar ama diğer algoritmalar da desteklenebilir. MD5 hash çok zayıf görülmektedir. Yakın bir zamanda uygulamalar AES-128-CMAC temelinde mevcut olacak ve kullanıcıların mümkün olduğu zaman bunları kullanma konusunda teşvik edilecektir.[42]

Autokey otomatik anahtar yönetimi ve NTP serverlarının doğrulanması için 2010 yılında tanımlanmıştır. Ancak güvenlik araştırmacıları Autokey protokolünde bu protokolü kullanışsız kılan bazı zafiyetler belirlemişlerdir. [40]

AutoKey dizi katmanı public anahtar şifrelemesi ve dijital imzaların kullanılması önerilmediğinden ve MAC ile yapılan korumada MD5 algoritmasının kullanılmasından dolayı yaptığımız çalışmada SSL sertifikalarında RSA veya ECDSA algoritması kullanıldığı için MD5' e algoritmasına göre daha güçlü güvenlik sağlamaktadır.

Autokey'in ikamesi olarak görülen Network Time Security (NTS) adlı bir çalışma devam etmektedir. Temmuz 2018 itibariyle 12 numaralı taslak "Çalışma Grubu Son Çağrı" evresinde olup. Okuyucularının bu mekanizmaları benimsemesi tavsiye edilmektedir. [39]

Bölüm 3

NTP

3.1 NTP (Network Time Protocol)

NTP İnternetin en eski protokollerinden biridir[1]. Değişken gecikmeli ağ yolları üzerinden iletişim kuran, güvenilir olmayan bilgisayar sistemleri arasında zamanı senkronize etmek için tasarlanmıştır.

NTP' nin en yaygın çalışma şekli hiyerarşik olarak istemci/sunucu biçimidir. İstemciler, bir dizi önceden yapılandırılmış ve genellikle zaman içinde statik olan sunucudan zamanlama bilgisi talep etmek için sorgular gönderir. NTP hem simetrik hem de asimetrik kriptografik kimlik doğrulamayı desteklerken [2], pratikte bu çalışma modları nadiren kullanılır.

NTP' nin amacı bir istemcinin saatini UTC zamanı ile senkronize etmesini sağlamak ve bunu yüksek bir doğruluk derecesi ve yüksek bir stabilite derecesi ile yapmaktır. Bir WAN kapsamında, NTP daha az milisaniyelik bir hassasiyet sağlayacaktır. Ağ kapsamı arttıkça, Global Positioning System (GPS) alıcısı veya caesium oscillator gibi hassas bir zaman kaynağı kullanılarak NTP'nin doğruluğu artabilir. LAN'larda alt milisaniye hassasiyetine ve alt mikrosaniye hassasiyetine izin verir.

3.2 Ağ Zaman Protokolü, Zaman ve Zaman Tutma

NTP'yi ele almak için zamanın ölçümü konusunun kendisinin dikkate alınması gerekmektedir. Bu bağlamda aşağıdaki bazı zaman ölçümü terimlerini ele almak yararlı olacaktır:

- Stability: Bir saatin sabit frekansı ne kadar başarılı şekilde sürdürebildiğini ifade eder.
- Accuracy: Saatin frekansı ve mutlak değerinin, standart referans zamanla ne kadar uyumlu olduğunu ifade eder.
- Precision: Saatin doğruluğunun belirli bir süre tutumu sisteminde ne kadar devam ettirilebildiğini ifade eder.
- Offset: İki saat arasında mutlak zaman farkını ifade eder.
- Skew: Zaman içinde ofset farklılığını ifade eder (zaman içinde ofsetin birinci dereceden türevi).
- Drift: Zaman içinde sapma değişkenliği (zaman içinde ofsetin ikinci dereceden türevi).

NTP, bir bilgisayarın süre ölçümündeki üç kritik metriğine ilişkin farkındalığını sağlamak için tasarlanmıştır: Yerel saatin seçili referans saate göre ofseti, yerel bilgisayar ile seçili saat sunucusu arasındaki ağda geliş-gidiş gecikmesi ve referans saatle ilişkili olarak yerel saatin maksimum hatasının bir ölçümü olan yerel saat dispersiyonu. Bu bileşenlerin her birisi NTP'de ayrı olarak tutulurlar. Sadece ofset ve gecikme ölçümlerinin doğruluğunu sağlamaz, aynı zamanda yerel saatin referans saat sinyali ile senkronizasyonu sağlamasına ve senkronizasyon sürecinin maksimum hata sınırlarının belirlenmesine olanak tanır ki bu sayede kullanıcı ara yüzü sadece zamanı değil, aynı zamanda zaman kalitesini de belirleyebilir.

3.3 Evrensel Zaman Standartları

Zamanın sadece zaman olduğu düşünülebilir ancak konu bu değil. Evrensel Zaman referans standardının farklı versiyonları bulunmaktadır ancak bu standartlardan iki tanesi ağ zaman tutumu için dikkate alınmaktadır.

UT1, Evrensel Zamanın temel ilkesidir. Her ne kadar kavramsal olarak anlamı 0 ° boylamdaki Ortalama Solar Zaman olsa da Güneş'in kesin ölçümü zordur. Bu nedenle de hesaplama uzun tabanlı interferometri ile uzak gök cismi gözlemleri, Ay ve yapay uyduların lazer ölçümleri ve GPS uydu yörüngelerinin belirlenmesi üzerinden gerçekleştirilir. UT1 Dünya'nın her noktasında aynıdır ve Dünyanın özellikle Uluslararası Göksel Referans Çerçeve (ICRF) olmak üzere uzak gökcisimlerine kıyasla dönüş açısına orantılıdır ve bazı küçük ayarlamalar ihmal edilmektedir.

Bu gözlemler Dünyanın ICRF'e kıyasla açısının belirlenmesine olanak tanır ki buna Dünya Dönüş Açısı (ERA) adı verilir ve Greenwich Ortalama Yıldız Zamanı'na modern bir alternatiftir. UT1'nin aşağıdaki ilişkiyi takip etmesi gerekmektedir:

$$\text{ERA} = 2\Pi(0.7790572732640 + 1.00273781191135448\text{T}\mu) \text{ radians}$$
$$\text{Burada T}\mu = \text{(Julian UT1 date - 2451545.0 olarak kabul edilir)}$$

Koordine Evrensel Zaman (UTC) ise UT1'e benzer bir atomik zaman ölçeğidir. Günlük zamanın dayandığı uluslararası standarttır. Uluslararası Atomik Zaman (TAI) ile uyumlu olarak SI saniyeler işler. Günde genellikle 86,400 SI saniye vardır ancak artık saniye atlamaları dikkate alındığında 0.9 UT1 saniye sınırında tutulmaktadır. 2012 itibariyle bu atlamalar hep pozitif olmuş, bir gün 86.401 saniye haline gelmiştir [3].

NTP Greenwich Ortalama Zamanı'ndan (GMT) farklı olarak referans saat standardı olarak UTC'yi kullanmaktadır. UTC, TAI zaman standardını kullanmaktadır ki bu da temel haldeki iki çok ince düzey arasındaki geçişte sezyum-133 atomunun yaydığı radyasyonun 9.192.631.770 periyodunun 1 saniye kabul edilmesine dayanır ve UTC'nin kendisi gibi NTP'nin zaman zaman artık saniye ayarlamaları içerdiğini gösterir.

NTP bir "mutlak" zaman protokolüdür, bu nedenle de lokal zaman dilimleri - ve mutlak zamanın Dünya yüzeyindeki belirli bir lokasyona referansla takvim tarihine ve zamanına dönüştürülmesi - NTP protokolünün içsel bir parçası değildir. UTC'den yerel tarih ve saat olarak ifade edilen zamana dönüşüm yerel hizmet bilgisayarına bırakılmaktadır.

3.4 Sunucu ve İstemciler

NTP sunucu ve istemci kavramlarını kullanmaktadır. Sunucu zaman bilgisinin kaynağı, istemci ise kendi saatini sunucu ile senkronize etmeye çalışan bir sistemdir.

Sunucular birincil sunucu ya da ikincil sunucu olabilirler. Birincil sunucu (bazen telefon ağı referans mimarisinin terminolojisi kullanılarak stratum 1 sunucu olarak da adlandırılmaktadır) UTC zaman sinyalini direkt olarak - günümüzde oldukça yaygın bir şekilde - GPS sinyal kaynağı ya da konfigüre atomik saat gibi yetkili saat kaynağından alır. İkincil sunucu ise zaman sinyalini bir ya da daha fazla üst sunucudan alır ve kendi zaman sinyalini bir ya da daha fazla alt sunucu ve istemciye dağıtır. İkincil sunucular saat sinyal tekrarlayıcıları olarak düşünülebilirler ve rolleri de birincil sunucularda istemci sorgu yükünü azaltmak ve istemcilerine birincil sunuculara eşdeğer kalitede saat sinyali iletmektir. İkincil sunucuların yukarı ve aşağı yönlü olarak doğru bir hiyerarşiyle düzenlenmesi gerekir ve bu süreçte genellikle stratum terminolojisi kullanılır.

Daha önce belirtildiği üzere, stratum 1 sunucusu zaman sinyalini UTC referans kaynağından almaktadır. Stratum 2 sunucusu ise zaman sinyalini Stratum 1 sunucusundan alırken Stratum 3 sunucusu Stratum 2'den almaktadır. Stratum n sunucusu ise referans saat sinyalini stratum n - 1 sunucularından alır. Zaman sunucuları dizisi içinde senkronizasyon döngüsünü engellemek için bu stratum çerçevesi kullanılmaktadır.

İç saatlerini NTP zaman sinyali ile senkronize etmek için istemciler sunucular ile eşleşirler.

3.5 NTP Protokolü

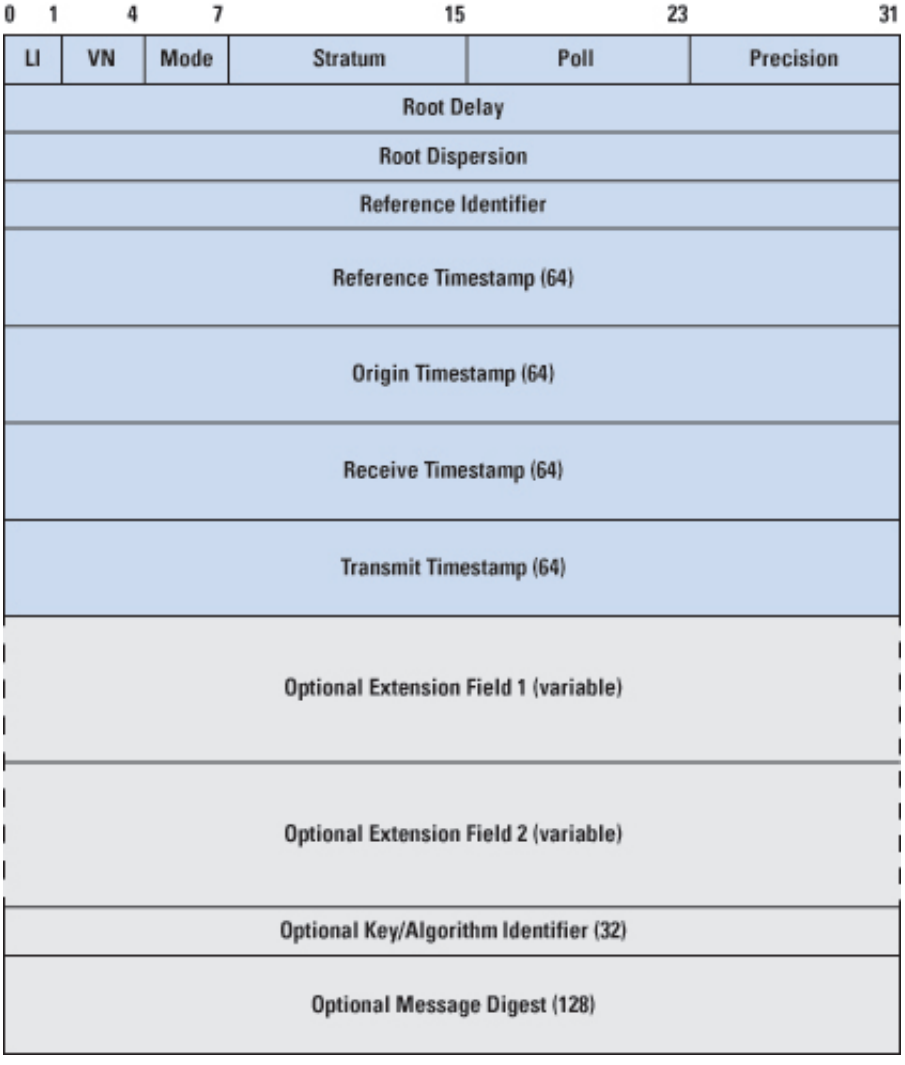
En basit haliyle NTP protokol, bir saat talep işlemidir ki burada istemci mevcut saati sunucudan talep eder ve kendi zamanını da bu talep ile birlikte iletir. Sunucu ise bu veri paketine kendi zamanını ekler ve paketi istemciye geri gönderir. İstemci paketi aldığında iki önemli bilgi edinmiş olur. Birincisi sunucudaki "referans zaman". İkincisi sinyalin istemciden geçerek sunucuya geri dönmesi için geçen ve yerel saat ile ölçülen "geçen zaman". Bu prosedürlerin tekrarlı iterasyonları lokal istemcinin ağ gecikmesinin etkilerini ortadan kaldırmasına ve bu sayede yerel saat ile sunucunun referans saat standardı arasındaki gecikme için stabil bir değer yakalamasına olanak tanır. Bu değer ise yerel saati ayarlamak ve sunucu ile senkronize etmek için kullanılır. Bu protokol değiş-tokuşunun daha ileri iterasyonları ise yerel istemcinin sürekli olarak lokal saati düzeltmesi ve lokal saat sapmasını önlemesine olanak tanır.

NTP Kullanıcı Datagram Protokolü (UDP) üzerinden çalışır. NTP sunucusu 123 port üzerinde istemci NTP paketini izler. NTP sunucusunu durum bilgisi yoktur ve alınan

paketlere alanlar ekleyerek ve önceki NTP işlemlerine herhangi bir referans olmaksızın paketi orijinal göndericiye geri göndererek teslim alınan her bir istemci NTP paketine basit bir işlemle yanıt verir.

İstemci NTP paketinin alınmasının ardından alıcı tarafından pakete olabildiğince çabuk şekilde sunucunun paket birleştirme mantığı dâhilinde zaman damgası eklenir. Bu paket daha sonra NTP sunucu sürecine iletilir. Bu süreç paketteki IP Başlık Adresi ve Port Alanlarını birbiriyle değiştirir, lokal saat değerlerini kullanarak NTP paketindeki pek çok alanın üzerine yazar, paketin çıkış zaman damgasını ekler, sağlama işlemini yeniden hesaplar ve daha sonra paketi istemciye geri gönderir.

İstemci tarafından sunucuya gönderilen NTP paketleri ve sunucudan istemciye iletilen yanıtlar Şekil 3.1' de görüldüğü şekilde ortak bir format kullanırlar.



Şekil 3.1: NTP Mesaj Formatı

NTP mesajının başlık alanları aşağıdaki tabloda belirtilmiştir:

Tablo 3.1: NTP Mesaj Baklış Alanları

LI	Artık Göstergesi (2 bit) Bu alan mevcut günün son dakikasının artık saniye içerip içermediğini gösterir. Bu alandaki değerler aşağıdaki gibidir: 0: Herhangi bir artık saniye uyarlaması yok. 1: Günün son dakikası 61 saniye 2: Günün son dakikası 59 saniye 3: Saat senkronize edilmiyor.
VN	NTP Versiyon Numarası (3 bit) (Mevcut versiyon 4).
MODE	NTP paket modu (3 bit) Mode alanının değerleri aşağıdaki gibidir: 0: Rezerve 1: Simetrik aktif 2: Simetrik pasif 3: İstemci 4: Sunucu 5: Yayın 6: NTP kontrol mesajı 7: Özel kullanım için rezerve
Stratum	Yerel zaman kaynağının stratum düzeyi (8 bit) Stratum alanının değerleri aşağıdaki gibidir: 0: Belirtilmemiş ya da geçersiz 1: Birincil sunucu 2-15: İkincil sunucu 16: Senkronize değil 17-255: Rezerve
Poll	Yoklama aralığı (8-bit imzalı tamsayı) Ardışık NTP mesajları arasındaki maksimum aralığın saniye cinsinden $\log_2$ değeri.
Precision	Saat doğruluğu (8-bit imzalı tamsayı) Sistem saatinin doğruluğu, $\log_2$ saniye cinsinden.
Root Delay	Sunucudan birincil referans kaynağa toplam gidiş-geliş gecikmesi. Bu değer 32-bit imzalı, saniye cinsinden sabit nokta sayıdır, kırılma noktası 15 ve 16 numaralı bitler arasındadır. Bu alan sadece sunucu mesajlarında önemlidir.
Root Dispersion	Saat frekans toleransından kaynaklanan maksimum hata. Bu değer 32-bit imzalı, saniye cinsinden sabit nokta sayıdır, kırılma noktası 15 ve 16 numaralı bitler arasındadır. Bu alan sadece sunucu mesajlarında önemlidir.
Reference Identifier	Stratum 1 sunucuları için bu değer harici referans kaynağını tanımlayan 4 haneli ASCII kodudur. (Bkz. Şekil 2.2) İkincil sunucular için ise senkronizasyon kaynağının 32-bit IPv4 adresi ya da senkronizasyon kaynağının IPv6 adresinin İleti Özet Algoritması (MD5) karmasının ilk 32 bitidir.

Code	External Reference Source
LOCL	uncalibrated local clock
CESM	calibrated Cesium clock
RBDM	calibrated Rubidium clock
PPS	calibrated quartz clock or other pulse-per-second source
IRIG	Inter-Range Instrumentation Group
ACTS	NIST telephone modem service
USNO	USNO telephone modem service
PTB	PTB (Germany) telephone modem service
TDF	Allouis (France) Radio 164 kHz
DCF	Mainflingen (Germany) Radio 77.5 kHz
MSF	Rugby (UK) Radio 60 kHz
WWV	Ft. Collins (US) Radio 2.5, 5, 10, 15, 20 MHz
WWVB	Boulder (US) Radio 60 kHz
WWVH	Kauai Hawaii (US) Radio 2.5, 5, 10, 15 MHz
CHU	Ottawa (Canada) Radio 3330, 7335, 14670 kHz
LORC	LORAN-C radionavigation system
OMEG	OMEGA radionavigation system
GPS	Global Positioning Service

Şekil 3.2: Referans Tanımlama Kodları (RFC 4330'den alınma)

Sonraki dört alan 64 bitlik zaman damgası değerini kullanır. Bu değer imzalanmamış 32 bitlik saniye değeri ve 32 bitlik fraksiyonel kısımdır. Bu örnekte 2.5 değeri 64 bitlik dizide şu şekilde ifade edilecektir:

0000|0000|0000|0000|0000|0000|0000|0010.|1000|0000|0000|0000|0000|0000|0000|0000

Zaman birimi saniyedir ve 1 Ocak 1900 tarihinden başlar ki bu NTP zamanının 2036 yılında döngüye gireceği anlamına gelmektedir (2038'deki 32 bit Unix zaman döngüsünden iki yıl önce).

Bu formatta ifade edilebilecek en küçük zaman fraksiyonu 232 piko saniyedir.

Reference Timestamp	Bu alan sistem saatinin en son ayarlandığı ya da düzeltildiği zamandır, 64 bitlik zaman damgası formatındadır.
Originate Timestamp	Bu değer talebin istemciden sunucu yönünde ayrıldığı zamanı gösterir, 64 bitlik zaman damgası formatındadır.
Receive Timestamp	Bu değer istemcinin talebinin sunucuya iletildiği zamanı gösterir, 64 bitlik zaman damgası formatındadır.
Transmit Timestamp	Bu değer sunucu yanıtının sunucudan ayrıldığı zamanı gösterir, 64 bitlik zaman damgası formatındadır.

Protokolün temel işleyişinde istemci sunucuya bir paket gönderir ve paketin istemciden ayrıldığı zamanı Origin Timestamp alanına kaydeder (T1). Sunucu ise paketin sunucu tarafından alındığı zamanı kaydeder (T2). Yanıt paketi ise orijinal Origin Timestamp ve

paketin alım zamanı olan Receive Timestamp ile yeniden birleştirilir ve daha sonrasında da mesajın istemciye geri gönderildiği zaman olarak Transmit Timestamp ayarlanır (T3). İstemci de paketin alındığı zamanı (T4) kaydeder ve şekil 3.3'de görüldüğü üzere istemcide dört zaman ölçümü gerçekleştirilir.

Timestamp Name	ID	When Generated
Originate Timestamp	T1	time request sent by client
Receive Timestamp	T2	time request received by server
Transmit Timestamp	T3	time reply sent by server
Destination Timestamp	T4	time reply received by client

Şekil 3.3: NTP İşlem Zaman Damgaları (RFC 4330'den alınma)

Bu dört parametre, İstemci üzerinde Zaman Tutma bölümde ele alınacak olan saat senkronizasyon işlevinin yerine getirilmesi için istemci zaman tutumu fonksiyonuna iletilir.

Opsiyonel Anahtar ve İleti özeti alanları istemci ve sunucu tarafından 128 bitlik gizli anahtarı paylaşmasına ve paylaşılan bu gizli anahtarı kullanılarak NTP mesaj alanlarının ve anahtarın 128 bitlik MD5 karmasının oluşturulmasına olanak tanır. Bu yapı da istemcinin man-in-the-middle saldırılarında hatalı yanıtların enjekte edilmesini tespit edebilmelerini sağlar.

Protokolün işleyişine dair son kısım yoklama sıklığı algoritmasıdır. NTP istemcisi düzenli aralıklarla NTP sunucusuna mesaj iletir. Bu düzenli aralık genellikle 16 saniye olarak ayarlanır. Eğer sunucuya ulaşılamıyorsa NTP bu yoklama aralığından çekilecek, her bir başarısız yoklama denemesi için geri çekilme zamanını ikiye katlayacak ve bunu yoklama aralığı 36 saat olana kadar devam ettirecektir. NTP sunucu ile yeniden senkronizasyon sağlamaya çalıştığında yoklama sıklığı artacak ve 2 saniye aralıklarla 8 paketlik bir gönderim yapacaktır.

İstemci saati sunucu saatine kıyasla yeterli ofsetle çalıştığında NTP bu yoklama aralığını genişletecek ve her 4 ila 8 dakikada (ya da 256 ila 512 saniyede) sekiz paket iletecektir.

3.6 İstemci Üzerinde Zaman Tutma

NTP işleyişinin bir sonraki kısmı istemci tarafında NTP sürecinin yerel saat için sunucuya yapılan periyodik yoklamalar tarafından ortaya konan bilgiyi nasıl kullandığıdır.

NTP yoklama işleminde istemci ile sunucu arasındaki gecikme istemci tarafından tahmin edilebilir. Şekil 3.3'te görülen zaman alanları kullanılarak iletim gecikmesi yoklamanın iletilmesi ile yanıtın alınması arasındaki toplam zamandan sunucunun yoklamayı işleme koyması ve yanıtlaması arasındaki farkın çıkarılması ile hesaplanabilir:

$$\delta = (T4 - T1) - (T3 - T2)$$

İstemci saatinin sunucu saatine göre ofseti de aşağıdaki şekilde tahmin edilebilir:

$$\theta = \frac{1}{2}[(T2 - T1) - (T3 - T4)]$$

Dikkat edilmesi gereken nokta bu hesaplamanın istemciden sunucuya ağ yolu gecikmesinin sunucudan istemciye yol gecikmesi ile aynı olduğunu varsaymasıdır.

NTP son sekiz ölçüm gecikmesinin en düşüğünü $\delta\theta$ olarak alır. Seçili ofset θ_0 ise en düşük günde yapılan ölçümdür. θ_0 ,δ_0 değerleri NTP güncelleme değeri halini alır.

Bir tek sunucu ile bir istemci konfigüre edildiğinde istemci saati, sunucu ofset değeri kabul edilebilir aralıkta olduğu sürece sunucu saati ile aradaki ofset sıfır olana kadar döngüsel bir operasyon ile ayarlanır.

Çok sayıda sunucu ile istemci konfigüre edildiğinde ise istemci adaylar arasında senkronizasyon için tercih edilen sunucuyu seçmek için bir seçim algoritması kullanır. Zaman sinyallerinin kümelenmesi uç değerdeki sunucuların dışlanması ile gerçekleştirilir ve daha sonra da algoritma en düşük stratum ve minimum ofset ve sapma değerleri olan sunucuyu seçer. Bu işlem sırasında NTP tarafından kullanılan algoritma ise Marzullo Algoritması'dır[4].

NTP istemci tarafında konfigüre edildiğinde istemci saatini referans zaman standardı ile senkronize tutmaya çalışır. Bunu yerine getirebilmek için ise NTP lokal zamanı konvansiyonel olarak küçük ofsetlerle ayarlar (daha büyük ofsetler devam eden işlemlerde sorunlara neden olabilir, tıpkı artık saniyenin işlenmesinde olduğu gibi). Küçük ayarlamalar adjtime() sistem çağrısı ile gerçekleştirilir ki bu da zaman doğrulaması gerçekleştirilene kadar yazılım saati sıklığını değiştirerek saati değiştirerek gerçekleştirilir. Saatin değiştirilmesi büyük zaman ofsetleri için yavaş bir süreçtir: Tipik değişim saniyede 0.5 ms'dir.

Bu konu NTP Ekosistemi Bölümünde detaylı olarak anlatılacaktır.

3.7 NTP Çalışma Modları

Sistem xntpd ve aşağıdaki metodlar kullanılarak diğer sunucularla senkronize edilebilir:

- İstemci Modu
- Broadcast Modu
- Multicast Modu
- Simetrik Aktif Mod

3.7.1 İstemci Modu

İstemci modunda sunucu diğer sunuculardan zamanı almak için kontroller yapar. Tüm sunucular kontrol edildikten sonra yerel ana makine hangi sunucuyla senkronize olacağını seçer. Sunucuyu istemci modda ayarlamak için NTP konfigürasyonunda kontrol edilecek sunucun adının ve ip adresinin bulunduğu bir sunucu bilgisi olmalıdır.

3.7.2 Broadcast ve Multicast Modu

Bu modda sunucu kontrolü yapılmaz, yerel ağdaki NTP paketleri dinlenir. NTP konfigürasyonunu bu modlarda ayarlamak için konfigurasyon dosyasına "Broadcast Yes" veya "Multicast Yes" yazılması gerekmektedir.

3.8 Basit Bir NTP Konfigürasyon Dosyası

Basit bir ntp konfigurasyon dosyası örneği:

```
# --- GENERAL CONFIGURATION ---
server  aaa.bbb.ccc.ddd
server  127.127.1.0
fudge   127.127.1.0 stratum 10

# Drift file.

driftfile /etc/ntp/drift
```

Şekil 3.4: NTP Konfigurasyon Dosyası

Ntp servisin düzgün çalışıp çalışmadığını kontrol etmek için birkaç tool kullanılabilir.

ntpq -p komutu ile o anki zaman durumu kontrol edilir.

ntpdc -c loopinfo komutı ile sunucuya yapılan en son bağlantıdan itibaren sistem saatinin ne kadarlık bir hata yaptığını saniye cinsinden göstermektedir.

ntpdc -c kerninfo komutu ile o anki kalan doğrulamayı gösterir.

ntpdate -d komutu ile NTP sunucusuna bağlanılır ve zaman sapması varsa bu gösterilir fakat sistem saati ayarlanmaz.

ntptrace "ntp sunucusu" komutu ile sistem saatin senkronizasyonu görüntülenebilir.

ntpdate "ntp sunucusu" Sistem saatinin belirtilen ntp sunucuna göre ayarlanmasını sağlar.

3.9 NTP Ekosistemi

NTP, DNS veya BGP gibi diğer temel internet protokollerinden daha akıcı bir şekilde gelişmiştir. NTP, RFC 5905'te tanımlandığı zaman [1] protokol son on yılda sıkça değişen NTP referans uygulaması ntpd tarafından belirlenir [5]. örneğin, root distance Λ temel bir NTP parametresidir ancak RFC 5905'te farklı tanımlanmıştır [1].

3.9.1 NTP Protokolünün Arka Planı

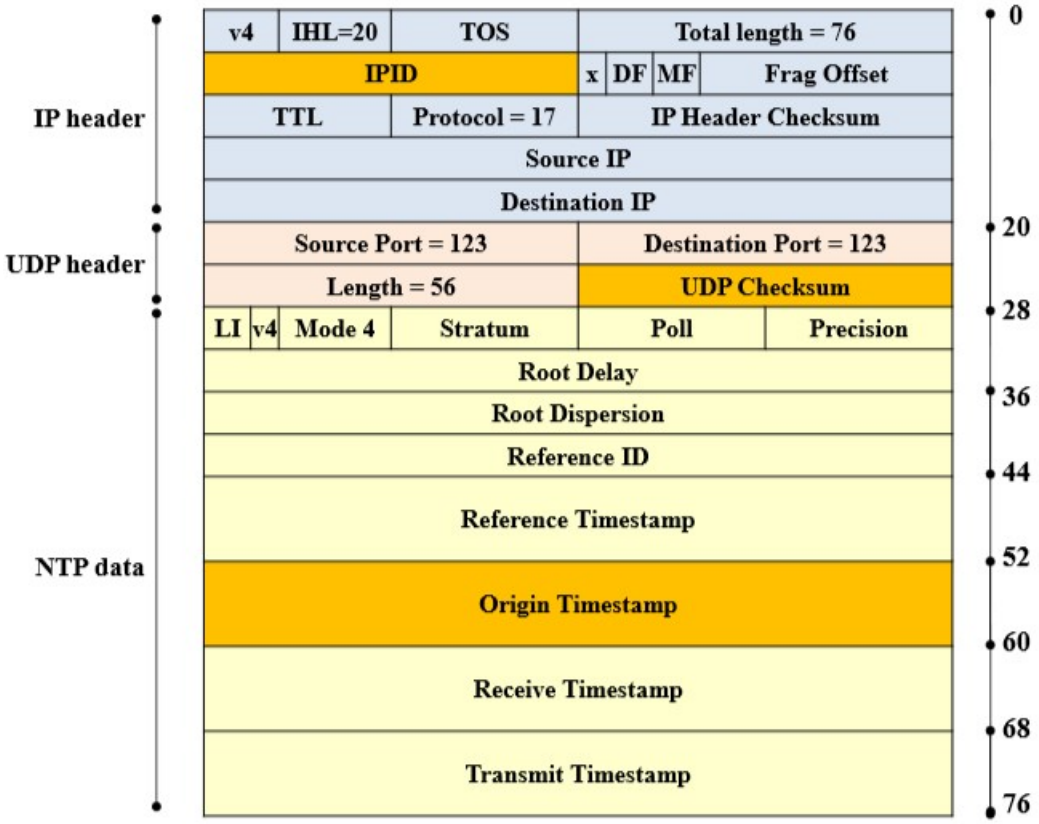
NTP en yaygın olarak hiyerarşik bir istemci-sunucu tarzında çalışır.[1] İstemciler bir dizi sunucudan zamanlama bilgisi talep etmek için sorgular gönderir. Bu sunucu grubu, istemci başlatılmadan ve zaman içinde statik kalmadan önce manuel olarak yapılandırılır. Genel olarak ntpd istemcisi 10 sunucuya kadar yapılandırılabilir.[2] 5 çevrimiçi kaynaklar, ntpd'yi tam olarak bir sunucuya (ör. time.apple.com) yüklemek için varsayılan olarak üç ila beş sunucu [6] ve belirli işletim sistemlerini (örneğin, MAC OS X 10.9.5) yapılandırmayı önerir. NTP hiyerarşisinin kökü, stratum2 istemci sistemlerine zaman bilgisi

[1] NTP aynı zamanda yayınlamayı da içeren bir dizi daha az popüler olan mod sağlar. Bu mod bir dizi istemcinin server'ı dinleyerek zaman bilgisini yayımladığı ve serverların (tipik olarak aynı stratumda olan) simetrik olarak değiş tokuş trafiğinde bulunduğu zaman alışverişini sağlar. Biz sadece istemci-sunucu modunu düşünüyoruz.

[2] örneğin, Temmuz 2015'te 14.04.1-Ubuntu'da NTP'yi kurarken, işletim sistemi önceden yapılandırılmış beş sunucuyla ntpd v4.2.6'yı kurardı.

sağlayan stratum1 NTP sunucularıdır. Stratum2 sistemleri, stratum 3 sistemlerine, vb., Stratum15'e kadar zaman sağlar. 0 ve 16 nolu stratum, sistemdeki asenkronizasyonu işaret eder. Stratum düşük olan NTP sunucuları genellikle internete büyük ölçüde zaman sağlar (örneğin, pool.ntp.org).

İstemci/sunucu iletişimin de bir NTP istemcisi ve sunucusu düzenli aralıklarla bir çift mesaj alışverişi yapar. İstemci sunucuya bir mesaj gönderir, mod 3, NTP sorgusu ile sunucu bir mod 4 NTP cevabını verir. Bu iki mesaj alışverişi şekil 2.5 'te gösterilen IPv4 paketini kullanır ve mod 4 yanıtında aşağıdaki dört önemli zaman damgasını uyarır:
T_1 Origin timestamp. İstemci mod 3 sorgusu gönderdiğinde istemcinin sistem zamanı.



Şekil 3.5: Mode 4 NTP Packeti Denkleri ve Sağlama Toplamlarını (checksum) Vurgulayarak

T_2 Receive timestamp. Sunucunun mod 3 sorgusu aldığı zaman sunucunun sistem zamanı.
T_3 Transmit timestamp. Sunucunun mod 4 yanıtı gönderdiğinde sunucunun sistem zamanı.
T_4 Destination timestamp. İstemci mod 4 yanıtı aldığı zaman istemcinin sistem zamanı (Paket içinde değil.)

Gidiş-dönüş gecikme δ değişim sırasında:

$$\delta = (T_4 - T_1) - (T_3 - T_2) \quad (1)$$

Offset θ Bir istemcinin saati ile bir sunucunun saati arasındaki zaman kaymasını ölçer. Forward (istemci -> sunucu) ve reverse (sunucu -> istemci) ağ yollarındaki gecikmelerin simetrik ve $\delta/2$'ye eşit olduğunu varsayalım. Ardından sunucu ve istemci saati arasındaki boşluğun mod 3 sorgusu için T_2 (T_1 + $\delta/2$) ve mod 4 yanıtı için (T_3 -(T_4 - $\delta/2$)). Bu iki miktarın ortalaması offseti verir:

$$\theta = \frac{1}{2}[(T_2 - T_1) - (T_3 - T_4)] \quad (2)$$

Bir NTP istemcisi, uyarlanabilir ve nadiren, zaman alacağı tek bir sunucuyu (önceden yapılandırılmış sunucu kümesinden) seçer. Seçilen sunucunun IPv4 adresi, bir sistemin gönderdiği her NTP paketinin referans kimliği alanına kaydedilir ve referans zaman damgası alanı, referans kimliğine en son senkronize edildiğinde kaydedilir. şuna dikkat edilmelidir ki herhangi bir sorgulama yapan ve senkronizasyon için kullanılan S2 server'ı IPv4 NTP sunucusunu kullanan S1 sunucusunu kesinlikle tanır.[3]

Infrequent clock updates. NTP istemcinin saatini nadiren güncellemektedir çünkü (1) istemci ve sunucunun, istemcinin saat disiplini algoritmaları sunucuya senkronize etmesinden önce sekiz ila yüzlerce mesaj alışverişi yapması gerekir [1] ve (2) mesajlar, rastgele bir yoklama işlemi tarafından uyarlamalı olarak seçilen, yoklama aralıklarında (dakikaların sırası ile) değiştirilir[1].

Authentication. İstemci saldırganla değil gerçek NTP sunucusuyla konuştuğunu nereden bilmektedir? NTPv4 desteklerken pratikte nadiren kullanılan hem simetrik hem de asimetrik şifreleme kimlik doğrulaması yapılmaktadır.

Tablo 3.2: En Yüksek NTPD Sürümleri rv Verisi, Mayıs 2015

Ntpd sürümü	4.1.1	4.2.6	4.1.0	4.2.4	4.2.0	4.2.7	4.2.8	4.2.5	4.4.2
Sunucular	1984571	702049	216431	132164	100689	38879	35647	20745	15901

Tablo 3.3: En Yüksek İşletim Sisitemi OSES rv Verisi, Mayıs 2015

İşletim Sisitemi	Unix	Cisco	Linux	BSD	Junos	Darwiv	Vmkernal	Windows
Sunucular	18209571	1602963	835729	38188	12779	3625	1994	1929

[3]128-bit IPv6 adresleri önce toplanır ve daha sonra 32 bit referans kimliği alanına kaydedilmeden önce kesilir [1]. Bu nedenle, bir IPv6 sunucusunu tanımlamak için sözlük saldırısına ihtiyaç duyulur.

Simetrik kripto grafik kimlik doğrulama, şekil 3.5'teki NTP paketine, NTP paket içeriğinin m simetrik anahtarı, k ile anahtarlanmış bir MD5 hash'ını MD5 ($k \parallel m$) [7] olarak ekler.[4] Simetrik anahtarın manuel olarak önceden yapılandırılmış olması gerekir; bu rastgele istemcilerden gelen sorguları kabul etmesi gereken ortak sunucular için bu çözümü oldukça hantal hale getirir. (NIST, önemli kamu stratum1 sunucularını çalıştırır ve simetrik anahtarları yalnızca yılda bir kez ABD posta veya faks yoluyla kaydeden kullanıcılara dağıtır [8]; ABD Deniz Kuvvetleri de benzer bir şey yapmaktadır [9]). Asimetrik şifreleme kimlik doğrulaması, RFC 5906 açıklanan Autokey protokolü tarafından sağlanır [2]. RFC 5906 standartlara uygun bir belge değildir ('Bilgilendirici' olarak sınıflandırılmıştır), NTP istemcileri varsayılan olarak Autokey protocol ilişkilendirmeleri istemez [11] ve birçok genel NTP sunucusu Autokey'yi desteklememektedir (örneğin, NIST zaman sunucuları [8], pool.ntp.org'daki birçok sunucu). Aslında, ntpd istemci sinin lider geliştiricisi 2015 yılında yazdığı [12]: "Kimse autokey kullanmamalı. Veya diğer yönden, eğer Autokey protocol kullanıyorsanız, kullanmayı bırakmalısınız."

3.9.2 NTP Ekosisteminin Ölçülmesi

Bu bölümde NTP ekosisteminin durumuna kısaca bakmaktayız. Ekosistemin ölçülmesi, NTP sunucularının IP adreslerini dağınık (wild) ortamlarda keşfederek başlar. Bir zmap çalştırılarak [13] mod 3 NTP kullanarak IPv4 adres alanının taranması, 12-22 Nisan 2015 tarihlerinde yapılan sorguları, 10.110.131 IP'den mod 4 yanıtı aldı. [5] Hangi IP'lerin NTP kontrol sorgularına cevap verdiğini belirlemek için haftalık taramalar yapan Ocak-Mayıs 2015 tarihine ait veriler. Veriler openNTPproject ile arttırıldı [14] . (Bu taramalar, kısa kontrol sorgularına cevap olarak büyük paketler gönderen potansiyel DDoS amplifikatörlerini tanımlamak için tasarlanmıştır.[15]). OpenNTPproject NTP'ye değişkeni oku (rv) kontrol sorguları gönderir ve verilen yanıtları günlüğe kaydeder. rv yanıtları aşağıdakiler de dahil olmak üzere yararlı bilgiler sağlar: Sunucunun işletim sistemi, ntpd sürümü, referans kimliği, ofset θ zamanı ve referans kimliğinin süresi arasında ve daha fazlası.

[4] MD5 ($k \parallel m$) bir şifreleme mesajı onay kodu (MAC) sağlaması amaçlanmıştır, ancak MD5 bu şekilde kullanıldığında MAC kesin olarak güvenli bir MAC değildir ve ayrıca uzunluk uzatma saldırılarına açıktır; HMAC yerine kullanılmalıdır [32]. Dahası, MD5 daha fazlası lehine amortismana tabi tutulmuştur. SHA-256 gibi güvenli karma işlevler [33].

[5] NTP kontrol sorgusu taramaları, 2014'ün bir parçası olarak [15]' 'nin araştırmasında birkaç' mega amplifikatör 'bulundu: milyonlarca yanıtla tek bir sorguya cevap veren NTP sunucuları. Bizim mod 3 taramamız da bunlardan buldu.

Dağınıklıkta (wild) işletim sistemleri ve istemciler. OpenNTPproject'in *rv* verilerini dağınık (wild) ortamdaki işletim sistemleri ve ntpd istemcileri hakkında bilgi sahibi olmak için kullanmaktadır. Önemli bir konu olarak, *rv* verisi tamamlanmamış; *rv* sorguları güvenlik duvarları ve diğer middleboxlar tarafından kesilebilir, NTP istemcileri bu sorguları reddedecek şekilde yapılandırılabilir ve *rv* cevapları bilgi içermez.

Tablo 3.4: En Yüksek Linux Kernellerin rv Verisi, Mayıs 2015

Kernel	2.6.18	2.4.23	2.6.32	2.4.20	2.6.19	2.4.18	2.6.27	2.6.36	2.2.13
Sunucular	1123780	108828	97168	90025	71581	68583	61301	45055	29550

Tablo 3.5: Dataset'te Stratum Dağılımı

Stratum	1	2	3	4	5	6	7-10	11-15
Sunucu	115351	1947771	5354911	1277941	615633	162161	218371	187348

İşletim sistemleri açısından, Tablo 3.3' de Unix, Cisco veya Linux çalıştıran birçok sunucuyu gösterir. Tablo 3.4 Linux çekirdeğinin yaygın olarak v2 olduğunu. Bu arada, tablo 3.2' de ntpd v4.1.1 (2001'de yayımlandı) ve v4.2.6'nın (2008'de yayımlandı) en popüler olduğunu görmekteyiz. Mevcut sürüm v4.2.8 (2014'de yayımlandı). Sonuç olarak, dağınık (wild) ortamda pek çok eski NTP sistemi bulunmaktadır. Bu nedenle, laboratuvar deneyleri ve saldırılarılar için iki NTP referans uygulamaları: ntpd v4.2.6p5 ve ntpd v4.2.8p2 kullanılmıştır.[34]

Bad timekeepers. Bad timekeepers - zaman bilgisi sağlamayn sunucular - dağınık (wild) ortamlarda görülür. Mod 3 sorgusuna yanıt veren her IP için θ (denklem (2)) ofsetini hesaplayıp, mod 3 sorgunun Ethernet frame zamanından T1, mod 4 sorgunun Ethernet frame zamanından T_1 alınıp ve mod 4 NTP yükünden T_2 ve T_3. Birçok bad timekeepers tespit edildi $-\theta \geq 10$ sn'de 1.7M, stratum 0 veya 16 da 3.2M idi ve her ikisinin birleştirilmesi toplamı verir 3.7M bad timekeepers.

Topoloji. Bir sistemin referans kimliği zaman aldığı sunucuyu gösterdiğinden, taramalar NTP'nin hiyerarşik istemci-sunucu topolojisinin bir alt kümesini oluşturulmasına olanak sağlanmıştır. Bununla birlikte, bir referans kimliği bir istemcinin önceden yapılandırılmış sunucuları yalnızca hakkında bilgi sağlar. Daha fazla bilgi edinmek için, 28-30 Haziran 2015 tarihlerinde, topolojide yalnızca bir ana sunucuya sahip olan her IP'ye ek bir mod 3 NTP sorgusu göndermek için nmap kullanılmıştır. Bu mevcut verilerle birleştirilerek potansiyel olarak NTP sunucularını çalıştıran toplam 13.076.290 IP ortaya çıkmıştır.

Ayrıca bad timekeepers işleyicilerine senkronize olan istemciler hakkında daha fazla bilgi edinmek için, 1 Temmuz 2015'te, θ>10 san. süresinde 1.7M sunucularının her birine bir liste sorgusu göndermek için openNTPproject'in tarama altyapısını kullanılmıştır. İzleme listesi yanıtları artık birçok sunucu tarafından devre dışı bırakılmaktadır, çünkü DDoS yükseltme saldırılarında kullanılmıştır. [15], 22,230 bad timekeepers' dan yanıtlar alındı. Monlist yanıtları, NTP paketleri (herhangi bir moddaki) sunucuya göndermiş olan tüm IP'leri listeleyen bir bilgi kaynağıdır. Her birinden yalnızca 3. ve 4. mod verilerinin çıkarılması monlist yanıtı ve mevcut verileri birleştirerek toplam 13.099.361 potansiyel NTP sunucusu tespit edildi.

Stratum. Tablo 3.5 Stratum'ların tüm veri kümelerindeki dağılımını göstermektedir. NTP istemcisi ile stratumlar arasında bire bir eşleme olmadığına dikkat edin; NTP istemcisi, çeşitli stratum sunucuları ile yapılandırılabildiğinden, istemci stratum'ı, seçtiği sunucuya bağlı olarak senkronizasyonu değişebilir.

Tablo 3.5 'de, stratum 3 en yaygın kullanılanı ve [15] birçok senkronize edilmemiş (stratum 0 veya 16) sunucu bulunmaktadır.

Degree distribution. Bir NAT veya güvenlik duvarının arkasındaki istemcilerle ilgili bilgilerin yanı sıra bir istemcinin yapılandırıldığı ancak senkronize edilmediği sunucuları hariç tutar.[6] Degree distribution oldukça eğridir. Veri setimizdeki 13.1M IP'lerin yaklaşık 3.7M'inde (%27.8) NTP hiyerarşisinde istemciler var. İstemcileri olan bu 3.7M sunuculardan %99.4'ü 10'dan az istemciye sahipken, yalnızca %0.2'si 100'den fazla istemciye sahip. Bununla birlikte, 100'den fazla istemciye sahip sunucular, her sunucuda en az $50,5$ istemciye sahip en iyi 50 sunucuya sahip olan, her sunucuda ortalama 1.5 K'nin üzerinde birçok istemciye sahip olma eğilimindedir. Bu önemli sunuculardan ödün vermek (veya trafiğini ele geçirmek) NTP ekosisteminin büyük alanlarını etkileyebilir.

[6] Daha önceki çalışmalar [16], [35] Topolojileri elde etmek için yaygın olarak devre dışı bırakılan monlist yanıtlarını kullandı.

Bölüm 4

Network Time Protocol Güvenlik Modeli

4.1 NTP Güvenlik Modeli

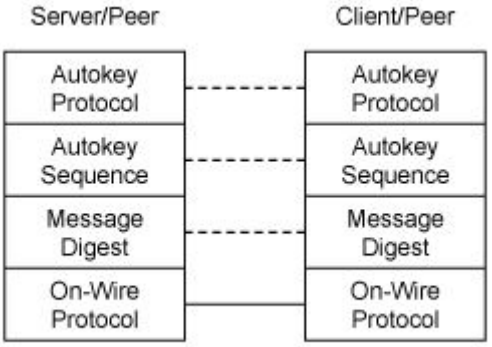
Saldırganlar büyük yanıt paketlerini (son DDoS Amplifikasyon Atağı), bazı zamanlı kritik hizmetleri sekteye uğratmak ve benzeri amaçlarla bu protokol ağı tıkamak için kullanabilmektedirler. NTP'de mümkün olan bazı saldırı türleri bulunmaktadır.

Bunlardan bazıları:

- Bir saldırganın, bir ya da daha fazla paketi yinelediği yineleme atağı.
- Man in the middle attack (MITM - iki nokta arasındaki bağlantının izinsiz bir şekilde izlenmesi), burada davetsiz misafir yetkili istemci ve sunucu arasındaki paketleri ele geçirebilmektedir.
- Delay attack, istemci ve sunucu arasındaki paketlerin belirli ya da değişken bir süre için yönlendirildiği ama değiştirilmediği yönlendirme saldırısı.
- Bir DDoS saldırısında, saldırgan savunmasız bir makine bulur, bunu bir botmaster'a çevirir ve diğer savunmasız sistemlere kötücül yazılımlar bulaştırır. NTP DDoS, saldırganın yanıltıcı (spoofed) SYN paketlerini ilettiği ve sunucu bu pakete

yanıt verdiğinde yanıtın direkt olarak SYN paketindeki yanıltıcı IP'ye gittiği bir reflektif DDoS saldırısı türüdür. DDoS'ta amplifikasyon faktörünü kullanarak saldırganlar saldırı sırasında trafik hacmini arttırırlar. Elde edilen sonuçlar bize bir NTP DDoS saldırısında 1GB bant genişliği olan bir saldırganın amplifikasyon faktörünü kullanarak neredeyse 250 GB'lik bir saldırı gerçekleştirebileceğini göstermektedir.

4.2 NTP Hiyerarşik Güvenlik Modeli



Şekil 4.1: NTP Güvenlik Modeli

NTP'deki güvenlik katmanları mümkün olan çeşitli saldırıları ve her bir katmanın ilgili saldırılara karşı nasıl bir koruma sağladığı aşağıdaki bölümlerde değinilmektedir.

4.2.1 On-Wire Protokolü

İstemci ile sunucu arasındaki paket transferi için kullanılan protokol. Kopya paketleri ve sahte paketleri tespit etmek için on-wire protokolü NTP paketinde 64 bitlik zaman bilgisi kullanır ki bir saldırganın bunu tahmin etmesi pek olası değildir.

Tüm protokol modlarında, iletilen zaman damgası önceki paketin iletilen zaman damgası ile eşleşirse duplicate paket tespit edilir. Bu durumda; yenilenen kopya paket, zaman damgası hesaplamaları üzerinde ilave bir etkisi olmadan atılır. Bu sayede bu katman tekrar atakları gibi ataklara karşı koruma sağlar. İstemci/sunucu ve simetrik modlarında; loopback (geri döngü) testi, istemci istek paketindeki iletilen zaman damgasını sunucu yanıt paketindeki asıl zaman damgası ile karşılaştırır. Bir tutarsızlığın tespit edilmesi, sunucu paketinin sahte olduğunu, eski bir kopya olduğunu veya iletimde kaybolduğunu

işaret eder. Simetrik modlar da, bu tutarsızlık protokolün yeniden başlatılmasından veya gönderilen paketlerin üst üste binmesinden kaynaklanabilir.

İstemci/sunucu ve simetrik modlarında loopback (geri döngü) testini kullanarak tüm wiretap saldırıları engellenebilir. Broadcast modlarında, loopback testi mümkün olmadığı için, on-wire protokol'u sahte paketleri veya eski kopyaları algılamaz. Ancak, bu senaryolara göre, paketin iletimi sırasında değişime uğramadığı ve sunucunun güvenilir olduğu, yani araya giren bir saldırgan tarafından Masquarade yapılmadığı varsayılmaktadır.

4.2.2 Messege Digest

İstemci ile sunucu arasındaki paketler gönderme zaman damgası hariç durdurulabilir veya değiştirilebilir. Sahte veya güvenliğini kaybetmiş bir router veya güvensiz bir sunucu, iletilen paketteki değişiklik yapabilmektedir.[18] Bu paketin yanlış sınıflandırılmasına veya hatalı zaman bilgisi olarak teslim edilmesine neden olabilir. Bu saldırılara karşı önlem olarak NTP, mesaj özetini hesaplamak amacıyla simetrik anahtar şifrelemesi kullanan bir messege digest katmanına sahiptir. Messege digest, NTP uzatma protokolü ile birlikte bir Message Authentication Code (MAC) eklentisi içeren ve bir gizli anahtar kullanan MD5 gibi algoritmalar kullanılarak hesaplanır. MAC 32 bit anahtar kimliğini takip eden bir "messege digest" ten oluşur. MD5 gibi bir algoritma daha sonra messege digest in ileti karıştırmasını hesaplar ve bunu NTP paket başlığı ile birleştirir. Paketler aktarılırken messege digest hesaplanır ve MAC içine eklenir. Paketler alındığında, messege digest, MAC içindeki messege digest ile karşılaştırılır ve paket sadece iki messege digest eşit olduğunda kabul edilir.

Messege digest anahtarının sadece hedef sunucu ve istemci arasında paylaşıldığı varsayımına dayanarak mesaj özeti katmanı, middleman veya masquarade saldırılarına karşı da başarılı bir savunma gerçekleştirmektedir.

4.2.3 Autokey Sequence Key

NTP paketlerinin orijinalliğini sağlamak için bir ortak anahtar şifrelemesi ve sadece sunucudan client'a verilen cevaplarda kullanılan dijital imzalar kullanılarak orijinalliğin sağlanması için autokey sequence katmanı kullanılır. İstemci-sunucu modunda sunucu

her istemciye benzersiz bir çerez dağıtır. Sunucu istemci çerezi autokey' in sunucunun ve istemcinin IP adresleri MD5 karması olarak sıfır anahtar kimliği ve sunucu çerezi olarak hesaplar. Bir istek aldığında sunucu şifreli istemci çerezini döndşrşr ve cevaplar sunucuya özel işaret anahtarı kullanılarak imzalanır. Diğer taraftaki istemci istemci çerezinin şifresini çözer ve sertifikada bulunan sunucu ortak anahtarını kullanarak doğrular. Daha sonraki istekler için, hem istemci hem de sunucu mesaj özetini hesaplar ve doğrular. Böylece saldırganın bir sunucu imzasına bağlı olan client çerezine sahip olana kadar bir paketin sahtesini oluşturamaz.

Fakat cookie snatcher olarak bilinen bir güvenlik açığını kullanan saldırgan istemci ve sunucu IP adreslerini öğrenmek için istemci çerez isteğini gizlice dinleyebilir. Saldırgan daha sonra kendi genel şifreleme anahtarını kullanarak bir istemci çerez isteği başlatır. İstemci çerezini kullanan saldırgan kendini gerçek bir sunucu gibi göstererek veya istemci tarafından kabul edilebilecek sahte paketler gönderebilir.

4.2.4 The Autokey Protocol Layer

Bu protokol katmanı sertifikaları ve kimlik anahtarlarını almak için kullanılır. Aşağıdaki gibi farklı anahtar çiftleri vardır:

- Host Key : Bu anahtarlar istemci çerezini şifrelemek için kullanılır.
- Sing Key : Bu anahtarlar sertifikalarda uzantı alanlarındaki imzaları teyit etmek için kullanılır. İmza anahtarlarını değiştirmek tüm sertifikaları değiştirme ihtiyacını doğuracaktır.
- Identity Key : Bu anahtarlar masquerade attacks (maskeleme saldırıları)' dan korunmak için sing keys (imza anahtarlarını)' leri doğrulamada kullanılır.

4.3 Zaman Neden Önemli: NTP'ye Yapılan Bazı Saldırılar

NTP birçok sistemin arka planında gizlenmektedir. NTP sistemde başarısız olduğu zaman, sistemdeki birden fazla uygulama aynı anda başarısız olabilir. Bu tür başarısızlıklar yaşanmıştı. Örneğin, 19 Kasım 2012'de [31], iki önemli NTP (stratum 1) sunucusu, tick.usno.navy.mil ve tock.usno.navy.mil, yaklaşık 12 yıl geriye gitti ve Active Directory

(AD) de dahil olmak üzere kimlik doğrulama sunucuları, PBX'ler ve yönlendiriciler [38] gibi çeşitli cihazlarda kesintilere neden oldu. Bireysel NTP istemcilerinin istismarları, Tablo 4.1'de özetlendiği gibi diğer saldırılar için bir yapı taşı görevi de görmektedir.

TLS Sertifikaları. Birkaç araştırmacı [7] [19], [17] NTP'nin güvenli şifreli ve kimliği doğrulanmış bağlantılar kurmak için kullanılan TLS sertifikalarının güvenliğini zayıflamak için kullanılabileceğini gözlemledi. İstemciyi aynı zamanda geri gönderen bir NTP saldırganı, ana makinenin, saldırganın hileli olarak verdiği (saldırganın bağlantının şifresini çözmesine izin veren) ve o zamandan beri iptal edilen sertifikaları kabul etmesine neden olabilir[1]. Örneğin, Comodo [20] 'da bir uzlaşma ile hacker'ların *.google.com gibi domainleri olan sahte sertifikalar kullanmasına izin verilen Mart 2011'e ya da 2014 ortalarına kadar geri gidebilirsiniz. Alternatif olarak, saldırgan, istemciyi anahtar şifresi zayıf bir sertifikanın hala geçerli olduğu bir zamana geri gönderebilir. (örneğin, Heninger [21] İnternet'in TLS sunucularının $\%0,50$'ine özel anahtarlar elde etmek için anahtar üretiminde entropi problemlerinden yararlanılan 2012'ye ya da Debian OpenSSL'deki bir bug'ın, sadece entropinin 15-17 bitlik kısmının anahtarları için binlerce sertifika yayımlanmasına neden olan 2008'e [28]). Ayrıca, günümüzde çoğu tarayıcı 1024 bitlik RSA anahtarları için (root olmayan) sertifikalar kabul etse de kaynaklar iyi finanse edilen rakipler tarafından kırılabileceğini göstermektedir[22].

Tablo 4.1: Çeşitli Uygulamalar İle Yapılan NTP Saldırıları

Saldırı	Zaman Değişimi
TLS Certs	Yıllar
HSTS[23]	Bir Yıl
DNSSEC	Aylar
HPKP[24]	Aylar
DNS Caches	Günler
Routing (RPKI)	Günler
Bitcoin[25]	Saatler
API authentication	Dakika
Kerberos	Dakika

Bu saldırıların bazıları Selvi [17] tarafından gösterilmiştir.

DNSSEC. DNSSEC, Alan Adı Sistemi (DNS) verilerinin şifreleme kimliğini sağlar. NTP, "mutlak" DNSSEC doğrulaması gerçekleştiren bir DNS çözümleyicisine saldırmak için

[1]Saldırganın ayrıca sertifika iptal mekanizmalarını atlatması gerekir, birkaç araştırmacı [10], [36], [1] bunun çeşitli ortamlarda yapmak için nispeten kolay olduğuna dikkat çekmiştir. örneğin, bazı büyük tarayıcılar OCSP'ye güveniyor [37] bir sertifikanın iptal edilip edilmediğini ve varsayılan olarak "soft-fail" olarak kontrol etmektedir, ör. OCSP sunucusuna bağlanamadıklarında sertifikayı geçerli olarak kabul etmektedir. NTP tabanlı önbellek kullanımı, istemcinin görmüş olabileceği eski sertifika iptal listelerini (CRL) 'unutmasını' sağlayarak istemciye de yardımcı olabilir.

kullanılabilir, Örnek olarak şifreleme DNSSEC doğrulamasını geçemeyen sorgulara yanıtları döndüremez. Zamanında bir alan adı çözümleyicisi gönderen bir NTP saldırısı, DNSSEC şifreleme anahtarlarındaki ve imzalarındaki tüm zaman damgalarının süresinin dolmasına neden olur (DNSSEC'deki bölge imzalama anahtarları için önerilen ömür 1 aydır [24]); Çözümleyici ve tüm istemcileri, DNSSEC ile güvence altına alınan herhangi bir domain'e olan bağlantısını böylelikle kaybeder. Alternatif olarak, zaman içinde alan adı çözümleyicisini geri gönderen bir NTP saldırısı DNSSEC tekrarlama saldırılarına izin verir. Örneğin saldırgan, domain adı için belirli bir DNSSEC kaydının bulunmadığı ve çözümleyicinin o domaine bağlantısını kaybetmesine neden olan bir zamana gelir. DNSSEC imzaları için önerilen ömür 30 günden fazla olmadığından [24], bu saldırının çözümleyiciyi zaman içinde aylık olarak geri göndermesi gerekir (Eğer DNSSEC kaydı geçmişte daha ileri bir zamanda bulunmuyorsa daha fazla sürer.) .

Önbellek temizleme saldırıları: NTP, önbellekleri temizlemek için kullanılabilir. DNS veya örnek, bir çözümleyicinin genel bir ad sunucusuna yaptığı DNS sorgularının sayısını en aza indirmek için ön bellekleme kullanır, böylece ağ trafiğini sınırlar. DNS önbellek girişleri tipik olarak 24 saat kadar çalışır, bu nedenle bir çözücüyü gün geçtikçe ileri doğru ötelemek, önbellek girişlerinin çoğunun süresinin dolmasına neden olur [19], [7]. Yaygın bir NTP hatası (Kasım 2012'deki gibi), birden fazla çözücünün önbelleklerini aynı anda boşaltmasına neden olabilir ve aynı anda ağı DNS sorgularıyla doldurur.

Etki alanları arası yönlendirme. NTP, Kaynak Ortak Anahtar Altyapısından (RPKI) yararlanmak için kullanılabilir [25], bu BGP ile yönlendirmeyi güvence altına almak için yeni bir altyapıdır. RPKI, IP adres bloklarının ağlara tahsis edilmesini şifrelemek için rota kökenli yetkilendirmeleri (ROA'lar) kullanır. ROA'lar korsanların ağlarına tahsis edilmemiş IP adreslerine rota oluşturmalarını önler. Geçerli bir ROA eksikse, 'bağlı olan taraf' (yönlendirme kararları almak için RPKI'ye güvenen), eksik ROA'daki IP'lere olan bağlantıyı kaybedebilir. Bu nedenle, bağlı taraflar her zaman tam bir geçerli ROA seti indirmelidir; bunu yapmak için, şifreli imzalı 'manifest' dosyalarında listelenen tüm dosyaları indirdiklerini doğrulamalıdırlar. Bağlı olan tarafın, ROA'sı eksik olabilecek eski bir manifestoya geri dönmesini engellemek için, manifestolar monoton olarak artan "manifesto sayıları" na sahiptir ve genellikle bir gün içinde sona erer [26].

Kimlik Doğrulaması çeşitli servisler (örneğin, Amazon S3 [27], DropBox Core API'sı [23]), bir uygulamayı her sorguladığında doğrulama gerektiren API'ler sunar. Tekrar

atakları önlemek için, sorgular sunucunun yerel saatinin yazılı olduğu kısa bir pencerenin olduğu bir zaman damgası gerekir. Örneğin: [29]; mesela Amazon S3, 15 dakikalık bir pencere kullanmaktadır. Bir NTP saldırganı, istemcilerin dakikalar içinde zaman aşımına uğramış biletler sunmasını gerektiren Kerberos'ta tekrarlama saldırıları başlatabilir [30].

Bölüm 5

Tasarım ve Uygulama

5.1 NTP Paketlerine SSL Sertifika İmzasının Eklenmesi

Bu çalışmada aşağıdaki kapsamlardan yararlanılmıştır:

DNS sorguları için dnspython kütüphanesi kullanılmıştır. Dnspython, DNS'e hem yüksek hem de düşük düzeyde erişim sağlar. Üst düzey sınıflar, belirli bir ad, tür ve sınıftaki veriler için sorgular gerçekleştirir ve bir yanıt seti döndürür. Ntp serverden gelen ssl sertifika imzası eklenmiş ntp paketlerinin dns server üzerinde bulunan public sertifikalarını almak için kullanılmıştır.

Ntp paketlerin ssl sertfika imzası eklenip eklenmeyeceği belirlemek için ntpslip.py kütüphanesi kullanılmıştır. Bu kütüphane ntp isteklerinde client tarafı için NTP (RFC-1305) implementationu ve ntp ile ilgili fonksiyonları içermektedir. Sunucu tarafında gelen ntp paketlerin hangi modda geldiğine ilişkin olarak client tarfına cevap dönmektedir. Kütüphanede aşağıdaki gibi 8 tane mod kullanılmıştır.

```
MODE_TABLE = {
    0: "reserved",
    1: "symmetric active",
    2: "symmetric passive",
    3: "client",
    4: "server",
    5: "broadcast",
    6: "reserved for NTP control messages",
    7: "reserved for private use",
}
```

Şekil 5.1: Ntp Paket Modları (ntpslip.py için)

Çalışmamızda mod 7 ile gelen ntp isteklerine ssl sertifika imzasının eklenip client'a iletilmesini sağlamaktayız. Diğer moddlar da gelen ntp isteklerine ssl sertfikası eklenmeden ntp paketi gönderilmektedir. Ntp paketlerine eklenen ssl sertifka imzaları RSA veya ECDSA (Elliptic Curve Digital Signature Algorithm) algoritmasına göre oluşturulmaktadır. Ntp paketlerine ssl sertifika imzasının eklenmesi için geçen süre hesaplanarak pakete eklenmektedir.

```
    if self.mode == 7  and not self.is_request:
        start = time.time()
        self.ntpdata = packed[0:NTPPacket._PF_SIZE]
self.signature=""

packet_format=NTPPacket._PACKET_FORMAT_MODE_7_RSA
if self.is_ecdsa:
    print "Signing with ECDSA Private Key"
    self.signature = ecdsa_ops.sign_data(self.ntpdata)
    packet_format=NTPPacket._PACKET_FORMAT_MODE_7_ECDSA
else:
    print "Signing with RSA Private Key"
    self.signature = rsa_ops.sign_data(self.ntpdata)

        end = time.time()
        print "end-start:", end - start
        self.signingdelta = long((end - start) * 1000000)  # convert sec. to microsec.
        print self.tx_timestamp
        print "Signing took: ", str(end - start), " secs ", str(self.signingdelta), " microsecs."
        print "Signed: OK"
```

Şekil 5.2: Ssl Sertifika Eklenmesi İçin Geçen Zamanın Hesaplanması

Client tarafında ise ntp paketlerinin ssl sertifika imzasının çözümlenmesi için geçen zamanda hesaplanmaktadir.

```
        start = time.time()
sign_mech=None
        if len(self.signature) == rsa_ops._RSA_SIGN_LEN_:
    print "Signed with RSA Key..."
    sign_mech=rsa_ops
elif len(self.signature) == ecdsa_ops._ECDSA_SIGN_LEN_:
    print "Signed with ECDSA Key..."
    sign_mech=ecdsa_ops
if sign_mech:
                serverCert=dns_ops.getTXTRecord(self.hostname)
                certVerified = False
                signVerified = False
                if serverCert != "":
                        serverPublic = cert_ops.extract_public_key(serverCert)
                        certVerified = cert_ops.validate_cert(self.hostname, serverCert)
                else:
                        print "DNS Entry not found..."
    print "-------------"
                signVerified=sign_mech.verify_sign_with_public_key(self.signature,data[0:NTPPacket._PF_SIZE],
                serverPublic)
                if certVerified and signVerified:
                        print "Signature Check: OK Cert Verify: OK Length: OK"
                else:
                        print "Len: OK\nSignature Check: FAILED\nCert: ", certVerified, "\nSign: ", signVerified
        else:
                print "Len: FAIL Signature: FAIL"
        end=time.time()
    else:
```

Şekil 5.3: Ssl Sertifika İmzasının Çözümlenmesi İçin Geçen Zamanın Hesaplanması

Public sertifika parçalara bölünerek DNS üzerinde TXT kaydı olarak tutulmaktadır. Bu işlem için DKIM sisteminden yararlanılmıştır.[41]

ntpserver.mustafakogce.com.	60	IN	TXT	p1=MIIFZTCCBE2gAwIBAgISA5KxpFSgxFqeHRbyQrLdkaS5MA0GCSqGSIb3DQEBCwUAMEoxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MSMwIQYDVQQDExpMZXQncyBFbmNyeXB0IEF1dGhvcml0eSBYMzAeFw0xOTAyMTUxOTM4MDdaFw0xOTA1MTYxOTM4MDdaMCIxIDAeBgNVBAMTF3NzbG50cC5tdXN0YWZha29nY2UuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAjUXxnWG/DcRXv/PLWjDrUAY7kNqN1O/ND0siRHpiO7wEpQ8EgQ/4FSLyfWdgpsJqSzpWvFmbjG/kXVa2B40xXZH9P9LV4uYhn6Cb
ntpserver.mustafakogce.com.	60	IN	TXT	p2=TwOpjqZkvK8YUAUqISYius3YYzhjGqh06MSxmCm+14X7RkBl0Q1TJugqSeJlfhmddzCHIlXQh7lGv2l7INQSfKNW7OBNCkW9xgC2FgoWyxNpjrDZO9BwdVrR/cIBuIpxKpQC6qy0sZnfCCpgMM0nYzR2WiXLcZuGYqVIX6E/cmCHNfGyOO77TGvEyNevqFCIZsQ7ztL+2fZ6we2CgY1u47mQYo6Pes9uftG1q98pGah/YCyovwIDAQABo4ICazCCAmcwDgYDVR0PAQH/BAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBgNVHRMBAf8EAjAAMB0GA1UdDgQWBBSxkksVdsflMl9IpC/ccS6Ap62nlTAfBgNVHSMEGDAWgBSo

Şekil 5.4: DNS üzerindeki Public Sertifika Kaydı

```
C:\Program Files\ISC BIND 9\bin>dig ntpserver.mustafakogce.com -t TXT

; <<>> DiG 9.14.2 <<>> ntpserver.mustafakogce.com -t TXT
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51546
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ntpserver.mustafakogce.com.    IN      TXT

;; ANSWER SECTION:
ntpserver.mustafakogce.com. 60  IN      TXT     "p5=++ltj3A9IEBw+O83lnjDMIosBaTWtf9/RtyWoq3I/scegAT14WUY5ILfnnAayoGLPqkMyQJmemZ2huPdukwhi
VusVAHfY9g5pP38CynJWNnUTf8UWnldVCW9/RtHl0g/iby5XjqWl7E027e/vy/4WuzePRi/Q6B8VpGNT7Q0VuY+Ix1PU0m4/4Gbys+lZw5shDYRadDOGF0ACBzA3tKeqVu84djdje
IQyv6yy+azEAVPP8m0/WyLQbg="
ntpserver.mustafakogce.com. 60  IN      TXT     "p1=MIIFZTCCBE2gAwIBAgISA5KxpFSgxFqeHRbyQrLdkaS5MA0GCSqGSIb3DQEBCwUAMEoxCzAJBgNVBAYTAlVTM
RYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MSMwIQYDVQQDExpMZXQncyBFbmNyeXB0IEF1dGhvcml0eSBYMzAeFw0xOTAyMTUxOTM4MDdaFw0xOTA1MTYxOTM4MDdaMCIxIDAeBgNVBA
MTF3NzbG50cC5tdXN0YWZha29nY2Uu" "Y29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAjUXxnWG/DcRXv/PLWjDrUAY7kNqN1O/ND0siRHpiO7wEpQ8EgQ/4FSLy
fWdgpsJqSzpWvFmbjG/kXVa2B40xXZH9P9LV4uYhn6Cb"
ntpserver.mustafakogce.com. 60  IN      TXT     "p2=TwOpjqZkvK8YUAUqISYius3YYzhjGqh06MSxmCm+14X7RkBl0Q1TJugqSeJlfhmddzCHIlXQh7lGv2l7INQSf
KNW7OBNCkW9xgC2FgoWyxNpjrDZO9BwdVrR/cIBuIpxKpQC6qy0sZnfCCpgMM0nYzR2WiXLcZuGYqVIX6E/cmCHNfGyOO77TGvEyNevqFCIZsQ7ztL+2fZ6we2CgY1u47mQYo6Pes
9uftG1q98pGah/YCyovwIDAQABo4IC" "azCCAmcwDgYDVR0PAQH/BAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBgNVHRMBAf8EAjAAMB0GA1UdDgQWBBSx
kksVdsflMl9IpC/ccS6Ap62nlTAfBgNVHSMEGDAWgBSo"
ntpserver.mustafakogce.com. 60  IN      TXT     "p3=SmpjBH3duubRObemRWXv86jsoTBvBggrBgEFBQcBAQRjMGEwLgYIKwYBBQUHMAGGImh0dHA6Ly9vY3NwLmlud
C14My5sZXRzZW5jcnlwdC5vcmcwLwYIKwYBBQUHMAKGI2h0dHA6Ly9jZXJ0LmludC14My5sZXRzZW5jcnlwdC5vcmcvMCIGA1UdEQQbMBmCF3NzbG50cC5tdXN0YWZha29nY2UuY2
9tMEwGA1UdIARFMEMwCAYGZ4EMAQIB" "MDcGCysGAQQBgt8TAQEBMCgwJgYIKwYBBQUHAgEWGmh0dHA6Ly9jcHMubGV0c2VuY3J5cHQub3JnMIIBAwYKKwYBBAHWeQIEAgSB9ASB
8QDvAHYAdH7agzGtMxCRIZzOJU9CcMK//V5CIAjGNzV5"
ntpserver.mustafakogce.com. 60  IN      TXT     "p4=5hB7zFYAAAFo8uFwqAAABAMARzBFAiAcoQIL1gbaMBPNdLutyxSj6zkjeFd4osBbbEB9N3RvpgIhAOv2Za1n2
qj0ZU4Yu+DPBk7Q0oB/8jvDEphspe/3zbiuAHUAY/Lbzeg7zCzPC3KEJ1drM6SNYXePvXWmOLHHaFRL2I0AAAFo8uFxCwAABAMARjBEAiBiIcoKSKVnQ6IZa8GV7NuhxlefyFF6Mh
yuAtK03LykbQIgQDzrvyq3fnHuzk62" "+69oC54PfKTR3XxqHklHgC6wTecwDQYJKoZIhvcNAQELBQADggEBABFYhZd1lcfCXxYuq2dPT2VLLX3Iy/7gMnbGmeTV1kIi9RBBIfCT
o6jHy4TdAviI1QiGqxzFH+RqTVwc4um2tkj3NauJgRTM"

;; Query time: 308 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Thu Jun 13 21:27:12 Turkey Standard Time 2019
;; MSG SIZE  rcvd: 1987
```

Şekil 5.5: DNS üzerindeki Public Sertifika' nın Sorgulanması

SSL sertifikalarında RSA ve DSA algoritmalarını kullanacağız. Bundan dolayı aşağıda ki kısımda RSA ve DSA algoritmalarının özetleri verilmiştir.

5.2 Şifreleme Algoritmaları

Eliptik Eğri Dijital İmza Algoritması (DSA), Dijital İmza Algoritmasından oluşmaktadır, Dijital İmza Algoritması da ElGamal İmza Algoritmasının türevidir. Dijital imzada anahtar olusturma işlemi iki kısım şeklinde olmaktadır. İlk kısımda farklı kullanıcılar arasında paylaşılabilecek algoritma parametrelerinin seçimi, ikinci kısım ise tek bir kullanıcı için açık ve gizli anahtarın belirlenmesidir.

Parametrenin Oluşturulması:

- Bir kriptografik özet fonksiyonu H belirlenir.
- Anahtar uzunluğu olarak L ve N belirlenir. L'nin değeri 512 ile 1024'ün arasında 64'ün katı olmalıdır.
- N-bit uzunluğunda asal q belirlenir.
- $p-1$, q nun katı olan L-bit uzunluğunda asal p modülü belirlenir.
- Çarpımsal mertebe modulo p de q olacak olan bir g seçilir. Bu işlem h $(1 < h < p-1)$ değeri için $g = h^{(p-1)/q} \bmod p$ ile hesaplanır.

Oluşan algoritma parametreleri (p, q, g) kullanıcılar arasında paylaşılabilir.

Kullanıcı anahtarlarının oluşturulması :

Birinci kısımda oluşturulan parametreler ile tek bir kullanıcı için gizli ve açık anahtar üretilir:

- $0 < x < q$ arasında bir x seçilir.
- $y = g^x \bmod p$ hesaplanır.
- (p, q, g, y) açık anahtarı oluşturur . x ise özel anahtarı.

İmzalama işlemi:

Özet fonksiyonu H, mesaj m olan:

- $0 < k < q$ arasında m için rassal bir k üretilir.
- $r = (gk \bmod p) \bmod q$ hesaplanır.
- $r = 0$, ise işlem farklı bir k değeri ile yeniden başlatılır.
- $s = (k - 1(H(m) + x * r)) \bmod q$ hesaplanır. $s = 0$ ise işlem farklı bir k değeri ile yeniden başlatılır.
- İmza (r, s) olur.

İlk iki kısım kullanıcı anahtarı oluşturulması için. İmzalama işlemlerinde hesaplaması en zor adım modüler üs almadır.'k'nın değerinin modüler tersini hesaplamak için $^{-1} \bmod q$ ikinci en zor işlemdir. $k^{q-2} \bmod q$ hesaplamak için Genişletilmiş Öklid Algoritması ya da Fermat'nın son teoremi kullanılabilir.

Algoritmanın Doğrulanması:

- $0 < r < q$ veya $0 < s < q$ sağlanmıyorsa imza ret edilir.
- $w = s^{-1} \bmod q$ hesaplanır.
- $u1 = H(m) * w \bmod q$ hesaplanır.
- $u2 = r * w \bmod q$ hesaplanır.
- $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$ hesaplanır.
- $v = r$ sağlanırsa imza kabul edilir.

RSA, güvenligini tam sayıları çarpanlarına ayırmanın zorluguna dayandırmaktadır. RSA algoritması üç aşamadan oluşmaktadır. Bunlar anahtar üretilmesi, sifreleme işlemi ve sifrenin çözümlenmesidir.

Anahtarın Üretilmesi:

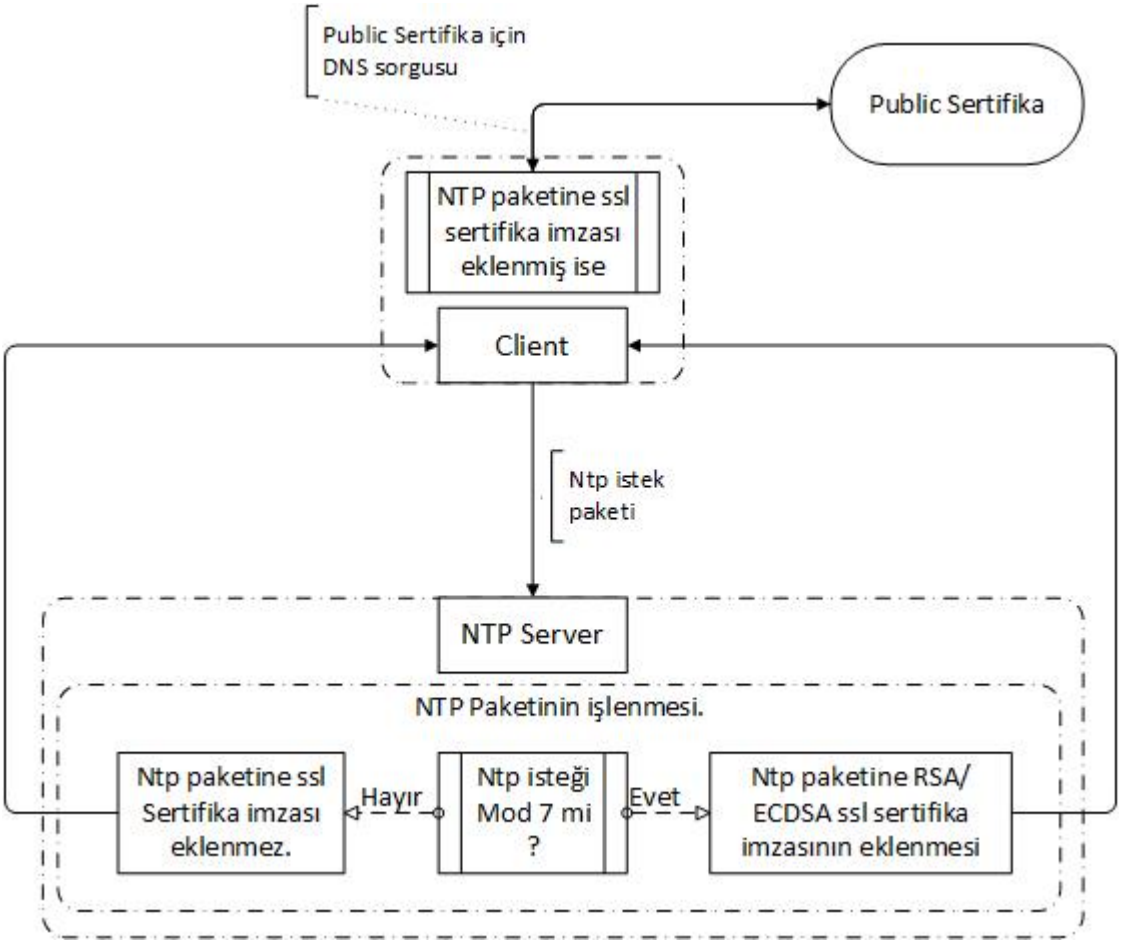
RSA' da bir ortak ve özel anahtar gerekmektedir. Ortak anahtar mesajı şifrelemek için kullanılır ve herkes tarafından bilinmektedir.Ortak anahtarla şifrelenen mesaj sadece özel anahtarla çözümlenebilir.

RSA' da anahtarın oluşumu:

- Birbirinden farklı iki asal sayı belirlenir, bunlar p ve q.
- $n = pq$ hesaplanır. n değeri açık ve cözel anahtar için mod değeri olacaktır.
- $\phi(n) = (p-1)(q-1)$ hesaplanır.
- e tam sayısı üretilir. Üretilen sayı, $1 < e < \phi(n)$ koşulunu sağlamalıdır. e değeri ortak anahtar olarak açıklanır.
- $de \equiv 1(\mathrm{mod}\phi(n))$ olacak şekilde bir d sayısı belirlenir. Bu değer özel anahtar olacaktır.

Ortak anahtar n 'den ve e 'den oluşmaktadır. Özel anahtar ise n 'den ve d 'den oluşur. (p,q ve $\phi(n)$ değerleri gizli olmalıdır çünkü bu değerler d 'yi hesaplama işleminde kullanılmaktadır.)

5.3 Çalışma Modeli

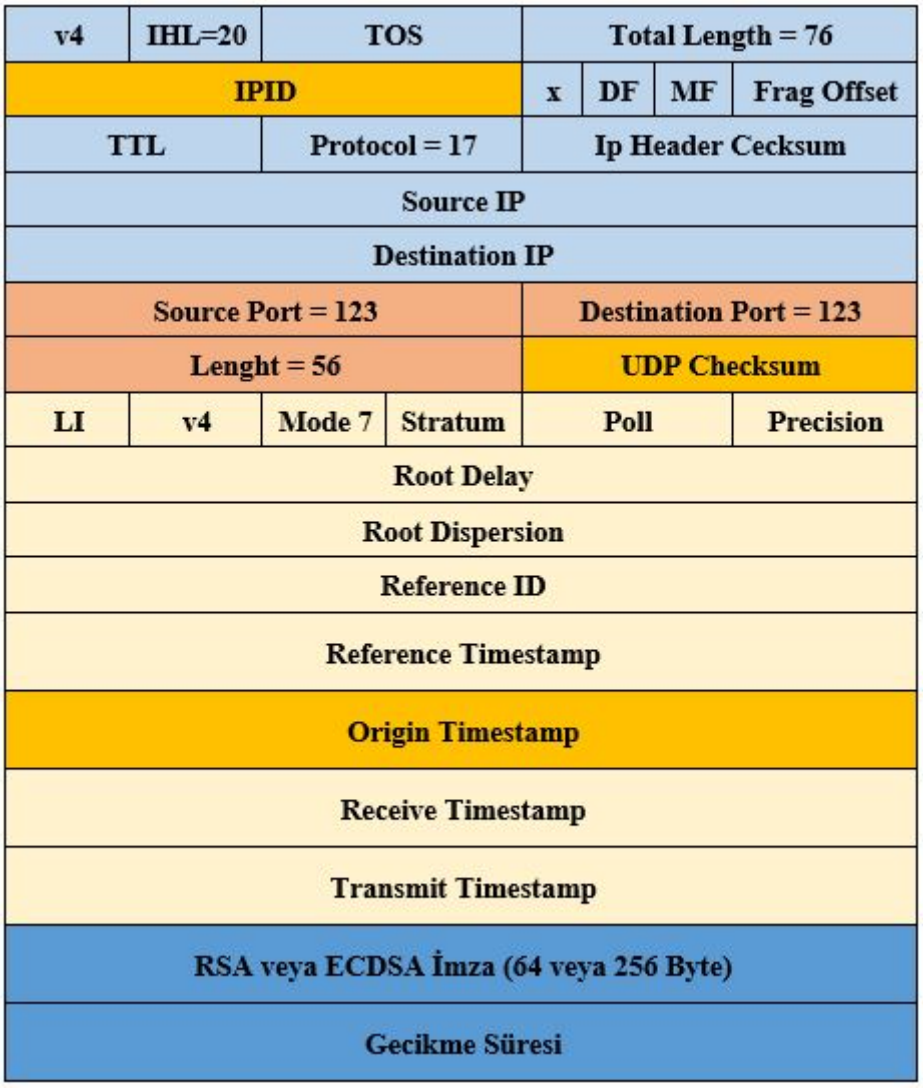


Şekil 5.6: Çalışma Modeli

Yerel ağdaki istemci bilgisayardan internet ortamına doğru yapılan ntp network trafiğini kendi yerel sunucumuza yönlendirerek trafiğin arasına girebilmekteyiz. Böylelikle yerel istemcilerin yapmış olduğu ntp isteklerine gelen cevapları yıllar mertebesinde değiştirebilmekteyiz. Yaptığımız çalışma modelinde bu tarz saldırıların engellenmesi planlanmaktadır.

Ntp sunucusu istemci tarafından gönderilen ntp istek paketinin özel mod (Mod 7) olarak kullanılması durumda isteği karşılayacak olan ntp sunucusunun ntp paketini ECDSA veya RSA algoritmalı bir ssl sertifika imzassı ekleyerek ve bu ekleme sürecinde geçen

zamanın hesaplanarak, micro saniye cinsinden paketlere eklenerek client' a gönderilmesi sağlanmaktadır. Şekil 5.6 da görüldüğü gibi gelen ntp istek paketi özel mod haricinde bir modda gönderilmiş ise ntp sunucusu istek hangi modda geldi ise ona uygun ntp paketi ile cevap verecek ve ssl sertifikası imzası eklenmesi işlemi gerçekleştirmeyecektir.



Şekil 5.7: Mod 7 SSL Sertifika İmzası Ekli NTP Paketi

Şekil 5.8' de Ntp' i paketine ssl sertifika imzasının eklenmesi ve bu işlem için geçen sürenin hesaplanarak pakte eklenmesini sağlayan kodu göstermektedir.

İstemci ise gelen ssl sertifika imzası ekli ntp paketinin çözümlenmesi için gerekli olan public sertifikayı almak için DNS sorgusu yapmaktadır. DNS sorgusuyla elde edilen public sertifika ile ntp paketi çözümlenerek zaman bilgisi ve sunucunun ssl sertifika imzası eklemesi için geçirdiği zaman bilgisi alınır. İstemci bu işlemler için geçen zamanı da

```
packed = struct.pack(packet_format,
     (self.leap << 6 | self.version << 3 | self.mode),
     self.stratum,
     self.poll,
     self.precision,
     _to_int(self.root_delay) << 16 | _to_frac(self.ro
     _to_int(self.root_dispersion) << 16 |
     _to_frac(self.root_dispersion, 16),
     self.ref_id,
     _to_int(self.ref_timestamp),
     _to_frac(self.ref_timestamp),
     _to_int(self.orig_timestamp),
     _to_frac(self.orig_timestamp),
     _to_int(self.recv_timestamp),
     _to_frac(self.recv_timestamp),
     _to_int(self.tx_timestamp),
     _to_frac(self.tx_timestamp),
     self.signature,
     _to_int(self.signingdelta))
```

Şekil 5.8: Ntp Paketine Geçikme Süresi ve SSL Sertifikanın Eklenmesi

hesaplamaktadır. Ntp paketinden elde ettiği sunucu ssl sertifika imzası ekleme zamanı ile kendisin çözümleme için harcadığı zamanı zaman bilgisine ekleyerek sonucu ekrana vermektedir.

Ts: Ntp sunucusunda ssl sertifika imzası ekleme için harcanan zaman bilgisi.

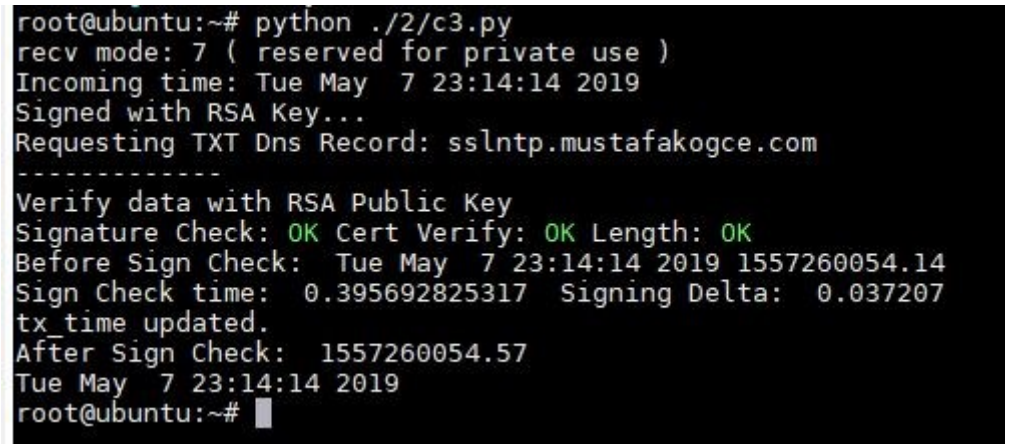
Tc : İstemcinin DNS sorgusunda harcadığı zaman ve ssl sertifika imzası ekli ntp paketinin çözümlenmesi için geçen zaman.

T : Ntp paketindeki gerçek zaman.

Gerçek Zaman : $T + Ts + Tc$

Sunucu ve istemci tarafındaki kodlarımız çalıştığı zaman aşağıdaki gibi bir çıktı almaktayız.

Şekil 5.9: Ntp Sunucu Tarafının Ekran Çıktısı



Şekil 5.10: İstemci Tarafının Ekran Çıktısı

Bölüm 6

Tartışma ve Sonuç

6.1 Tartışma ve Sonuç

NTP güvenlik modelinde, NTP paketi içerisindeki veriyi açık veriler olarak görür bu nedenle de kendiliğinden bu verileri değiştirmesi ya da şifrelenmesi için hiçbir işlemde bulunulmaz. Sadece kaynakların doğruluğunun teyit edilmesi ve ifade edilen saldırılardan sakınmaya çalışır. Saldırganların en belirgin hedefi zamana bağlı bazı kritik hizmetlerin devre dışı kalmasına neden olacak şekilde yanlış veya tutarsız zaman değerlerinin üretilmesi ya da protokolü maliyetli kripto grafik hesaplamalar gibi önemli kaynakları tüketmeye zorlamaktır.[18]

Çalışmamızda kullanılan SSL sertifika imzası ile maliyetli kripto grafik hesaplamaların önüne geçilmesi ön görülmüştür. Tasarımımız da ki maliyet sunucu tarafında ek imzalama süresi (Ts), istemci tarafında ek dns sorgusu ve imza kontrol süresidir (Tc).

Toplam maliyet Tt = Ts + Tc olmaktadır.

Saldırganın gerçekleştirebileceği saldırı türlerin dikkate alındığında. Sahte saldırıda saldırgan istemci veya sunucu tarafından kabul edilebilecek bir paket oluşturmaya çalışır. Wiretap saldırısında saldırgan network üzerinde dinlemesi yaparak istemci ve sunucu paketlerini kopyalar ve bunları sürekli olarak arşivleyebilir. Replay saldırısında ise saldırgan zaman protokulü paketlerin bir ya da bir kaçını yeniden gönderir. Duplicatede ise ağ zaman protokül paketin yeniden iletilmesinden dolayı gönderilen en son paketin yeniden dinletilmesini sağlar.

Bu aktif saldırıların yanı sıra paketlerin bozulma, sıranın taşması ya da sağlama toplamı hatasına neden olan bit hataları gibi nedenlerle kaybolduğu pasif ataklar da mevcuttur. NTPv4 spesifikasyonlarında tanımlanan azaltma algoritmalarının doğası nedeniyle küçük düzeyde yaşanan paket kayıpları bu algoritmaların performansını etkilemez. Fakat kayıp paketlerin protokol yeniden başlamasına neden olduğu bazı koşullar mevcuttur.

Saldırgan istemci veya sunucu paketlerinin sabit veya değişken bir zamanla iletilmesinin geciktirildiği ancak paket içeriğinin değiştirilmediği bir delay saldırısında bulunabilir. Eğer istemci ve sunucu arasındaki iki yöndeki gecikmeler büyük ölçüde aynı ise ofset hatası önemsiz olabilir. İki yöndeki gecikmeler önemli ölçüde farklı ise ofset hatası bu iki yönde gerçekleşen gecikmeler arasındaki farkın yarısıdır. Bu gibi saldırılar güvenliği ihlal edilmiş bir router ile gerçekleştirilmesi mümkün olabilir. [18]

Bir middleman saldırısında saldırgan araya girerek bir istemci veya sunucu paketini yakalayabilir ve bunun doğru bir şekilde iletilmesini engelleyebilir. Saldırgan daha sonra sunucu veya istemci tarafından kabul edilen sahte veya yanıltıcı paketler oluşturabilir. Bunlara alternatif olarak saldırgan bir AutoKey uzantı alanına eski ya da sahte bilgileri yerleştirmek için bir cut-and-paste saldırısı girişiminde bulunabilir. Ultimate saldırısın da saldırgan güvenilir bir sunucu gibi davranarak kendisini maskelemektedir.

Çalışmamızda kullanılan SSL sertifika imzalaması ile UDP protokolünün kullanımından kaynaklı ya da Middleman, Wiretap, Replay, Duplicate ve Sahte saldırı ataklarında ki ntp paket yapısının değiştirilmesinin önüne geçilmesi öngörülerek ntp paketinin bütünlüğünün sağlanması amaçlanmıştır.

NTP güvenlik çalışmalarında RSA ve DSA gibi şifreleme algoritmaları pahalı kabul edilir ancak MD5 ve SHA gibi hash algoritmaları farklı olarak değerlendirilir. [39]

Message Digest, AutoKey, RFC 5905, RCF 7384' de MD5 algoritması kullanılmaktadır. Bizim yağtığımız çalışmada RSA ve DSA algoritmaları kullanılarak MD5 algoritmasına göre daha güçlü bir güvenlik sağlanması amaçlanmıştır. Aşağıda ki şekilde MD5 algortiması ile RSA algoritmasının karşılaştırılması gösterilmiştir.

FACTORS	MD5	RSA
Key Length	64 bits, 128 bits , 256 bits , 512 bits	1024 bits , 2048 bits , 4096 bits
Block Size	128 bits	1024 bits
Developed	1992	1977
Cryptanalysis Resistance	Strong against Digital Certificate and very fast on 32 bit machines	Strong against Digital Certificate and data
Security	Secure	Secure but slow with large amounts of text
Rounds	4	1

Şekil 6.1: MD5 ve RSA Karşılaştırması

Ssl sertifkalarında RSA şifreleme algoritması yerine ECDSA kullanılması durumunda RSA ile çok uzun anahtarlarla sağlanan güvenlik ECDSA ile daha kısa anahtarla sağlanmış olacaktır.

Aşağıdaki şekilde RSA ve DSA key size'ların karşılaştırılması verilmiştir.

ECC(in bits)	RSA(in bits)
106	512
112	768
132	1024
160	2048
210	3072
283	7680
409	15360
571	21000

Şekil 6.2: DSA ve RSA Key Size Karşılaştırması

6.2 Gelecek çalışmalar

Çalışma sonunda hazırlanan kod kavram ispatı (Proof of consept, PoC) çalışması kapsamında olup, kodun daha hızlı çalışması için NTP' nin içerisine gömülmesi hedeflenmektedir.

Bölüm 7

Kaynakça

[1] D.Mills,J.Martin,J.Burbank,andW.Kasch. RFC5905:NetworkTime Protocol Version 4: Protocol and Algorithms Specification. Internet Engineering Task Force (IETF) 2010.

[2] B. Haberman and D. Mills. RFC 5906: Network Time Protocol Version 4: Autokey Specification. Internet Engineering Task Force (IETF), 2010.

[3] https://en.wikipedia.org/wiki/Universal

[4] K. A. Marzullo, "Maintaining the Time in a Distributed System: An Example of a Loosely-Coupled Distributed Service," Ph.D. dissertation, Stanford University, Department of Electrical Engineering, February 1984,

[5] H. Stenn. Securing the network time protocol. Communications of the ACM: ACM Queue, 13(1), 2015.

[6] B. Knowles. NTP support web: Section 5.3.3. upstream time server quantity, 2004.

[7] D. L. Mills. Computer Network Time Synchronization. CRC Press, 2nd edition, 2011.

[8] The NIST authenticated ntp service.

[9] DoD Customers : Authenticated NTP.

[10] K. Kiyawat. Do web browsers obey best practices when validating digital certificates? Master's thesis, Northeastern University, 2014.

[11] Autokey Configuration for NTP stable releases. The NTP Public Services Project.

[12] S. Rottger. Finding and exploiting ntpd vulnerabilities, 2015.

[13] Z. Durumeric, E. Wustrow, and J. A. Halderman. Zmap: Fast internetwide scanning and its security applications. In USENIX Security, pages 605-620. Citeseer, 2013.

[14] J. Mauch. openntpproject: NTP Scanning Project.

[15] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir. Taming the 800 pound gorilla: The rise and decline of NTP DDoS attacks. In Proceedings of the 2014 Internet Measurement Conference, pages 435-448. ACM, 2014

[16] N. Minar. A survey of the NTP network, 1999.

[17] J. Selvi. Breaking SSL using time synchronisation attacks. DEFCON'23, 2015.

[18] NTP Security Analysis. https://www.eecis.udel.edu/ mills/security.html

[19] J. Klein. Becoming a time lord - implications of attacking time sources. Shmoocon Firetalks 2013.

[20] Comodo. Fraud incident. https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html, March 2011

[21] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman. Mining your ps and qs: Detection of widespread weak keys in network devices. In USENIX Security Symposium, pages 205-220, 2012.

[22] E. Barker and A. Roginsky. Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. NIST Special Publication.

[23] DropBox. Core API. https://www.dropbox.com/developers/core/docs

[24] O. Kolkman, W. Mekking, and R. Gieben. RFC 6781: DNSSEC Operational Practices, Version 2. Internet Engineering Task Force (IETF), 2012.

[25] M. Lepinski and S. Kent. RFC 6480: An Infrastructure to Support Secure Internet Routing. Internet Engineering Task Force (IETF), 2012.

[26] E. Heilman, D. Cooper, L. Reyzin, and S. Goldberg. From the consent of the routed: Improving the transparency of the RPKI. ACM SIGCOMM'14, 2014.

[27] Amazon. Simple storage service (s3): Signing and authenticating rest requests.

[28] P. Eckersley and J. Burns. An observatory for the SSLiverse. DEFCON'18, July 2010.

[29] E. Hammer-Lahav. RFC 5849: The OAuth 1.0 Protocol. Internet Engineering Task Force (IETF), 2010.

[30] J. Kohl and C. Neuman. RFC 1510: The Kerberos Network Authentication Service (V5). Internet Engineering Task Force (IETF), 1993.

[31] L. Bicknell. NTP issues today. Outages Mailing List.

[32] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In Advances in CryptologyCRYPTO96, pages 1-15. Springer, 1996.

[33] S. Turner and L. Chen. RFC 6151:Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms. Internet Engineering Task Force (IETF), 2011.

[34] Aanchal Malhotra, Isaac E. Cohen, Erik Brakke, and Sharon Goldberg : Attacking the Network Time Protocol.

[35] C. D. Murta, P. R. Torres Jr, and P. Mohapatra. Characterizing quality of time and topology in a time synchronization network. In GLOBECOM, 2006.

[36] A. Langley. Revocation still doesn't work.

[37] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. RFC 6960: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol OCSP. Internet Engineering Task Force (IETF), 2013.

[38] M. Morowczynski. Did your active directory domain time just jump to the year 2000? Microsoft Server and Tools Blogs, November 2012.

[39] T. Mizrahi. Security Requirements of Time Protocols in Packet Switched Networks, 2014

[40] D. Reilly, Ed. Network Time Protocol Best Current Practices, 2012

[41] http://www.dkim.org/specs/rfc5585.html

[42] https://tools.ietf.org/html/rfc5905