2014 6th International Conference on Cyber Conflict
P.Brangetto, M.Maybaum, J.Stinissen (Eds.)
2014 © NATO CCD COE Publications, Tallinn

Detecting and Defeating Advanced Man-In-The-Middle Attacks against TLS

Enrique de la Hoz
Computer Engineering Department
University of Alcala
Alcalá de Henares, Spain
enrique.delahoz@uah.es

Rafael Paez-Reyes
Spanish Navy
Cyberdefence Area
Madrid, Spain
rpaerey@fn.mde.es

Gary Cochrane
Cyberdefence Area
Indra
Torrejon de Ardoz, Spain
gicochrane@indra.es

Ivan Marsa-Maestre
Computer Engineering Department
University of Alcala
Alcalá de Henares, Spain
ivan.marsa@uah.es

Jose Manuel Moreira-Lemus
Cyberdefence Area
Spanish Navy
Madrid, Spain
jmorlem@fn.mde.es

Bernardo Alarcos
Computer Engineering Department
University of Alcala
Alcalá de Henares, Spain
bernardo.alarcos@uah.es

Abstract: TLS is an essential building block for virtual private networks. A critical aspect for the security of TLS dialogs is authentication and key exchange, usually performed by means of certificates. An insecure key exchange can lead to a man-in-the-middle attack (MITM). Trust in certificates is generally achieved using Public Key Infrastructures (PKIs), which employ trusted certificate authorities (CAs) to establish certificate validity chains.

In the last years, a number of security concerns regarding PKI usage have arisen: certificates can be issued for entities in the Internet, regardless of its position in the CA hierarchy tree. This means that successful attacks on CAs have the potential to generate valid certificates enabling man-in-the-middle attacks. The possibility of malicious use of intermediate CAs to perform targeted attacks through ad-hoc certificates cannot be neglected and are extremely difficult to detect.

Current PKI infrastructure for TLS is prone to MITM attacks, and new mechanisms for detection and avoidance of those attacks are needed. IETF and other standardization bodies have launched several initiatives to enable the detection of "forged" certificates. Most of these

initiatives attempt to solve the existing problems by maintaining the current PKI model and using *certificate pinning*, which associates certificates and servers on use. These techniques have significant limitations, such as the need of a secure bootstrap procedure, or pinning requiring some host-by-host basis.

This study proposes an evolution from pinning-in-the-host to pinning-in-the-net, by enabling mechanisms to validate certificates as they travel through a given network. Certificates would be classified as trusted or not trusted as a result of cross-information obtained from different sources. This would result in early detection of suspicious certificates and would trigger mechanisms to defeat the attack; minimize its impact; and gather information on the attackers. Additionally, a more detailed and thorough analysis could be performed.

Keywords: *certificate-pinning schemes, MITM attacks retaliation, SDN, OpenFlow*

1. INTRODUCTION

TLS [1] is an essential building block for securing virtually every application layer protocol and has also been successfully used to secure virtual private networks. A critical aspect for the security of any TLS dialog is authentication and key exchange, usually performed by means of X.509 certificates. An insecure key exchange can lead to an active third party (i.e. an attacker) being able not only to eavesdrop, but also to intercept and insert traffic in the communication in order to alter the setup process for the secure channel inserting himself effectively "in-the-middle" of the communication, thus hindering confidentiality and integrity.

Ideally, key exchange should only occur when there is certainty about the authenticity of the certificates involved. Trust in certificates is generally achieved using Public Key Infrastructures (PKIs), which rely on trusted third parties (called certificate authorities, CAs) to establish certificate validity chains [2], which are called certification paths. A communicating party assumes a certificate as authentic if the signature of the certificate can be traced back through a valid certification path up to a trusted CA. This method for validating certificates is the de facto standard in the Internet, and has been regarded as secure for decades.

Although the Public Key Infrastructure using X.509 Certificates (PKIX) [2] is meant to avoid the occurrence of man-in-the-middle attacks on TLS, recent incidents have clearly shown the weaknesses of the classical PKI model. The public CA model allows any trusted CA to issue a certificate for any domain name. A single trusted CA that betrays this trust, either voluntarily or by being compromised, can undermine the security provided by any certificates used in TLS just by issuing a replacement certificate that contains a rogue key, that is, a key not corresponding to the entity identified in the certificate.

A number of security concerns regarding PKIX usage have arisen in the last years, and it is foreseen that more incidents are likely to occur in the following years [3]. A certificate authority can issue certificates for any entity of the Internet, regardless of its position in the CA hierarchy

tree. A Spanish CA, for instance, can issue a certificate for a US government website, and vice versa. This was coherent with the decentralized nature of the Internet (avoiding single points of failure), but has turned instead into an "any-point-of-failure" problem. A successful attack on any CA in the hierarchy allows the attacker to generate valid certificates for any host in the Internet which will be blindly accepted by most users, browsers and Internet applications, thus enabling effective man-in-the-middle attacks. These attacks are not theoretical, but have been found in the real world. Comodo CA issued in 2011 certificates for major websites such as Google, Yahoo, Mozilla and Skype to an Iranian hacker [4]. The DigiNotar CA in the Netherlands was also removed as a trusted CA in most major browsers after issuing a Google certificate to a third party. Whether these incidents are the result of sophisticated attacks or poor security policies is irrelevant. The fact is that countries cannot just rely on the security of their own PKI infrastructures (or that of their allies). NATO can usually audit its own CA infrastructures and ensure their security. However, security breaches in an external CA can also jeopardize NATO own security. In addition, the possibility of malicious use of intermediate CAs to perform targeted attacks through ad-hoc certificates cannot be neglected [5], and these attacks are extremely difficult to detect. These rogue certificates can be used in man-in-the-middle attacks, which will not be detected by conventional mechanisms for PKIX certification path validation and revocation checks.

2. RELATED WORK

Current PKIX infrastructure for TLS is prone to MITM attacks, which are usually consummated by the use of forged certificates or by manipulating certificate path validation. IETF and other standardization bodies have launched several initiatives to enable the detection of "forged" certificates. Most of the proposals focus on minimizing the impact of certificate misissuance while maintaining the current PKI model almost unchanged in order to ensure compatibility, usability and low-cost deployment.

DNS-Based Authentication of Named Entities (DANE) [6] is a proposal to extend the secure DNS infrastructure DNSSEC [7] to store and sign keys and certificates which are used by TLS, so that clients can use this information to increase the level of assurance they receive from the TLS handshake process. Thanks to the use of DNSSEC, clients can verify that DNS information was provided by the domain operator and not tampered with while in transit.

The rationale behind DANE is that given that the DNS administrator for a domain name is authorized to provide identifying information about his jurisdiction zone, he should be allowed to make an authoritative binding between the domain name and a certificate that might be used by a host at that domain name. According to this line of thinking, the proper place to hold this information is the DNS database, securing the binding with DNSSEC.

This binding is done by means of a certificate association. A security association is composed by the domain name where the server application runs and some information from the certificate used to identify this application. A certificate association can also define the combination of a trust anchor and a domain name. This certificate association is represented by the TLSA

DNS resource record [6], which is used to associate a TLS server certificate or public key with a domain name. DANE defines several use cases, which allows to apply this binding information either to End Entities (EE) or to define new trust anchors that should be used to perform certificate path validation. A domain name administrator can even issue certificates for a domain without involving a third-party CA. A thorough description of DANE use cases can be found in [8].

Security associations are protected via DNSSEC. Taking into account that the deployment of DNSSEC infrastructure is still incomplete, any global proposal for certificate verification cannot rest solely on DANE. Moreover, certificate validation procedures will use only PKIX checks when no DANE information is available. An active attacker who is able to divert user traffic could block DANE traffic, so that he can bypassed these additional verifications. Moreover, there are situations where DANE information could fail to get to the End Entity due to server errors or broken intermediaries that filter DNSSEC errors. Under these circumstances, the End Entity performing the validation could assume an attack is undergoing and terminate the connection, or it could dismiss the error and proceed. The latter would mean that blocking DNSSEC traffic could help to bypass the DANE-defined procedures. Thus, in order for DANE to be effectively used to prevent MITIM attacks, a deployment of DNSSEC in clients, servers, DNS infrastructure and intermediaries (i.e., to avoid DNSSEC information filtering) is required. Taking into account the traditional resilience of network operators and manufacturers, we cannot rely solely on DANE to provide the kind of path validation we are looking for in this work. Finally, the verification of a key would require several DNSSEC queries that would introduce an undesired latency, unaffordable in some cases, e.g., SIP, XMPP.

In the short term, the basic technique that has been proposed to deal with this problem is known as *certificate pinning*, and relies on associating hosts with their *expected* X.509 certificates or public keys. *Pinning* is a way for clients to obtain a greater level of assurance in server public keys. By pinning a trusted known certificate (or public key), clients can detect any change either in the certificate or in the public key submitted by any server as part of any future TLS handshake.

There are two main problems related to pinning techniques. The first one is related to the process of bootstrapping the trust procedures, how we decide which associations are established. These associations can be set the first encounter with the host in a *Trust-On-First-Use* basis (TOFU), or can be defined by a list that is shipped with the application. The second one is the need for maintenance of the secure associations database, which is the secure creation of new associations and the revocation of existing ones if needed. Currently, there the two main proposals for certificate pinning are the Trust Assertion for Certificate Keys (TACK) Internet Draft [9] and the Public Key Pinning Extension for HTTP [10] promoted by Google.

In TACK, clients are allowed to pin to a server-chosen signing key (TACK signing key, TSK), which will be used to sign server's TLS keys. Given that the actual TLS keys are not pinned, the site is able to deploy different certificates and keys on different servers, without having the clients to renew its pins. Also since pins are not based on CA keys, there is no need to trust in CAs. TACK also defines a mechanism to activate pins. As part of the TLS handshake, a client

could request a compliant TACK server to send its TSK public key and signature. Once a client has seen the same hostname-TSK pair multiple times, it could decide to activate a time-limited pin for that pair. By time-limiting the pins, the potential impact of a bad pinning decision is bounded. The specification also mentions that pins could be aggregated and shared through a trusted third party but without defining either the infrastructure or the protocols required. This proposal, while promising, is still in a very early stage and accordingly not suitable for use in a production environment.

Public Key Pinning Extension (PKPE) for HTTP is conceptually quite similar to TACK but here the pins get delivered via a HTTP header and, accordingly, can only be applied to HTTP servers. This proposal defines a new HTTP header to enable a web host to tell browsers which public key should be present in the web host's certificate in future TLS connections. We can see this as a way to bootstrap public key pinnings. Once pinned, when connecting to a web server, the client can easily do PKIX checks and also can verify that one of the pinned keys for that server is present. The main drawback of this and other similar pinning techniques is that they do not protect the user against man in the middle attacks during the first connection attempt to the server. Also, such a MITM attack would not be detected until an update in the associations could be deployed to the hosts. This leaves an insecurity window that can be as long as one month in the PKPE case. To minimize this risk, a static list of pins is usually deployed with software packages. For instance, a total of 300 static pins are provided with the Google Chromium browser.

Another proposed solution is the 'sovereign keys' project by the Electronic Frontier Foundation [11] (EFF). This solution that uses a "semi-centralized, verifiably append-only data structure" containing the keys and revocations. These keys can only be added when it is strongly verified that the domain belongs to the requesting party. A browser would, when connecting to a TLS service, lookup the certificate from this key-store. Similarly to Certificate Transparency the existence of an append-only log with all CA-issued certificates is assumed.

Finally, pinning techniques require some configuration in a host-by-host basis and do not ship with a pre-established and well-defined mechanism for sharing pin information, even under the same domain. Currently very few sites publish pins, which limits the applicability of the proposal but it is expected that this situation will change in the near future, fuelled by the support by Google. Unfortunately, it is short-term solution and its scope is limited to HTTP so it is unable to help preventing MITM attacks against any other protocol secured by TLS.

The problem of verifying the authenticity of a given certificate can be affected by additional circumstances other than the presence of a rogue CA. For instance, a hostname can map to different servers, each with a different certificate and different CA chains, due to their dependence on different jurisdictions. Also, it is possible for a CA chain to change at any time, and this is out of the control of the administrators of the site. There is a proposal called *certificate transparency* [12], which tries to make the certificates that a certain CA has issued auditable and easy to track. This would make it easier for a site administrator to keep track of any new certificate issued for its site, usually a clear indication of a potential security breach. Participating entities should publish all certificates they issue so that clients could check

whether a certificate received by a server has a proof of publication. If the client is not able to obtain a cryptographic proof of publication, this could mean that the certificate has been forged. Note that this kind of verification can be provided by means of DANE.

Once again, the effectiveness of this technique is limited by the degree of deployment of the proposals. Certificate transparency can detect forged certificates issued by participating CAs but has none detection capabilities regarding non-participating CAs. This is an especially significant limitation, since usually a server is not concerned about misissuance by its own CA, but about the others (see, for instance, the TURKTRUST case [5],), these others CAs are out of its control. Finally, there is an inherent limitation derived from the very nature of the PKI model. Since the security of the whole PKI is the security of the weakest CA, and that these weakest CAs are not likely to be part of this initiative, the expected security improvement cannot be very significant.

There is a whole set of proposals that try to detect MITM attacks taking advantage from the fact that this kind of attacks are usually targeted attacks, rather than global scale attacks. This means that the attacker attempts to fool a specific target into believing the authenticity of the issued rogue certificate or key, while the rest of the Internet users are unaffected by the attack. Therefore, the victim will be receiving a certificate, which is different to the one seen by other Internet users. The *Perspectives* [13] and *Convergence* projects, in order to establish the validity of a received certificate, query designated nodes distributed over Internet, which act as *notaries*. A notary maintains a database of known server certificates. After the reception of a certificate by the client, she can check against the notary's version and flag mismatches as possible attacks. Notaries introduce a reputation scheme into the standard validation process.

Depending on the opinions received from these notaries, the certificate gets accepted or rejected. In practice, this voting scheme could be used to override the information used from the CA model. However, the client still has to trust these nodes (so they may become a point of failure if compromised), and there is a dependence on a pre-existing infrastructure.

There are some interesting initiatives in the Internet for sensing and mining information about the existing certificates, which could be used to produce a more valuable evaluation of the validity of a certificate. The most prominent examples are the ICSI Certificate Notaries Service [14] and the EFF SSL Observatory [15]. The ICSI Certificate Notaries Service passively collects certificates at multiple independent Internet sites, aggregating them into a central database almost in real-time. The ICSI Notary provides a public DNS interface allowing a client to query its database with the SHA1 digest of a certificate that it would like to check. The currently inactive project Certificate Catalogue by Google offered a service quite similar to this. The idea of deploying a set of sensors to passively detecting certificates in transit in order to identify common and uncommon patterns is also one of the key points of our proposal but, in our case, the set of provided parameters is richer in order to be able to perform a multigroup classification. Other initiatives like Crossbear or EFF SSL Observatory actively scan the Internet either by querying the TLS-enabled servers or asking the users to submit the certificates that they see.

The DetecTor Project [16] reuses the notary idea, but making every client to act as their own notary. To do this, the authors propose to use the Tor network to connect to the server under evaluation for the sole purpose of checking which certificate is seen when contacted from a different network location. This is essentially the same idea proposed in [17] but extended not only to HTTP but also to every protocol.

While the standardization work is progressing at a satisfactory speed, challenges remain. There is no common agreement of the design constraints and the types of threats that are supposed to be mitigated. The threat landscape is constantly evolving and an agreement about what threats need to be address does not exist.

In order to be actually effective, a widespread deployment is required by all this initiatives. This deployment can imply client, server and additional infrastructures (e.g., DNS infrastructure for DANE or CAs for Certificate Transparency).

3. MIDAS: A DISTRIBUTED "PINNING-IN-THE-NET" APPROACH FOR AUTOMATED CERTIFICATE ASSESSMENT

The aforementioned proposals offer only partial solutions to the certificate assessment problem. They usually require full deployment of the initiative (given the 'weakest link in the chain' property of PKI). Finally, pinning needs to be performed in a host-by-host basis, which is hardly scalable.

This study proposes MIDAS (*Man-in-the-middle Distributed Assessment System*). MIDAS is an evolution from the pinning-in-the-host techniques to pinning-in-the-net techniques, by enabling mechanisms to validate certificates as they *travel through a given network*. Our idea is to classify certificates as trusted or not trusted as a result of cross-information obtained from different sources in an automated and distributed manner. While there have been some initiatives on using automated classification techniques for certificate assessment in the Internet [18, 19], they usually require centralized analysis of massive amounts of training data to become effective. While this large corpus of data including both legitimate and rogue certificates can be assumed as available for the Internet scenario, it is hardly achievable in internal NATO networks, which are not only more reduced in size and traffic, but also present a lower security incident rate than what we find in open networks. In the following, we propose an approach to pin certificates as they pass through the network, which takes advantage of collective intelligence techniques and does not require extensive training data.

A. Environmental Assumptions

Our approach assumes an environment of a secure internal NATO network. This implies a number of network segments compartmentalized by a set of (physical or virtual) switches and routers. It also implies the existence of several network management infrastructure elements.

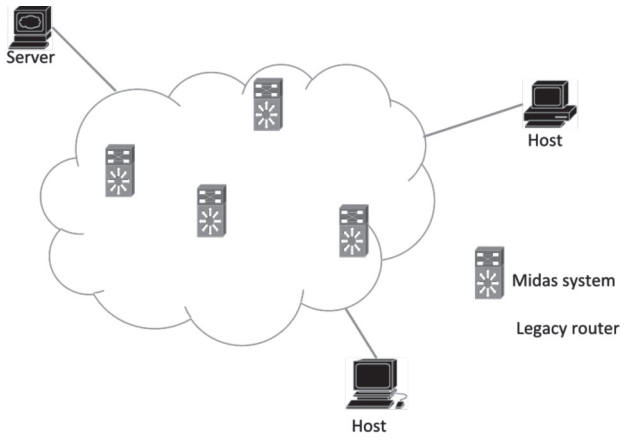
As for the threat model, we assume an insider attack scenario, since attacks from the outside will usually be handled using other techniques. We will assume the attacking entity to be an individual node or group of nodes, which are in minority with respect to the total nodes in the network. We will also assume that the targets of the MITM attacks (that is, the client and server between which the attackers intend to place themselves at) have not been completely isolated by the attackers (that is, both client and server are able to send data to other hosts in the network). We will later discuss techniques to ensure that these assumptions hold.

B. System architecture

Our system uses a distributed variation of the typical Intrusion Detection System Architecture [20], which encompasses the following elements:

- A distributed information source, consisting of a set of network probes. In our system, eventually any network element or host can act as a probe.
- A distributed analysis engine, which relies on Bayesian Networks to evaluate trust relationships according to information about the certificates involved and network history.
- A distributed reaction component, which allows to effective alter network topology in real time to transparently counter man in the middle attacks, thus ensuring network and service operation.

FIGURE 1: DISTRIBUTED MIDAS ARCHITECTURE



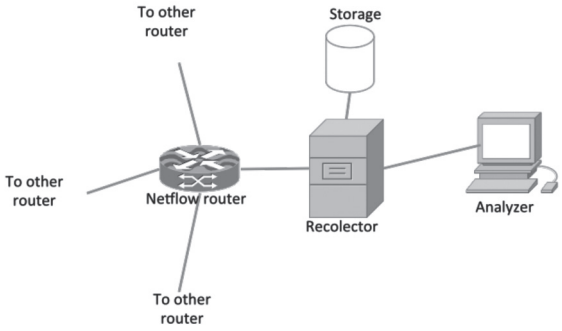
In this section we briefly outline each of these elements and their role in our proposal.

1) Network probes

As stated above, virtually any of the network elements within the communication infrastructure we want to secure may behave as a network probe. Basically, all that is needed is a network card that can act in promiscuous mode and capture packets from the network. Of course, such monitoring of traffic may have a serious impact in the performance of conventional hosts, so

we do not expect that all hosts in the network will act as probes. However, we assume that there are a sufficient number of probes distributed throughout the network. In particular, we rely on the existence of network management devices, which are specifically designed to be network probes. For instance, devices using NetFlow [21] or similar technologies for flow data analysis are especially suitable as probes in our system. These devices could, for instance, gather information about the TLS flows being established in the network, aggregating not only data about the certificates being used, but also the network path from source to destination or even the traffic patterns observed (e.g. an asymmetric flow with 80% of the traffic flowing from server to client).

FIGURE 2: ARCHITECTURE OF MIDAS SYSTEM



2) Distributed analysis engine based on bayesian networks

The MIDAS analysis engine, again, relies on distribution. Any node in the network can act as an analyzer, provided that it has information to analyze. Therefore, the most usual scenario is that network probes themselves act as analyzers in the case of probes residing in hosts, whereas Netflow collectors or analysis consoles act as analyzers in the case of probes residing in network management elements.

Analysis itself will be performed by using Bayesian Networks. A Bayesian network is a model that encodes probabilistic relationships among variables of interest. This technique is generally used for intrusion detection in combination with statistical schemes, a procedure that yields several advantages, including the capability of encoding interdependencies between variables and of predicting events, as well as the ability to incorporate both prior knowledge and data [22]. Each analyzer will have a built-in Bayesian network which is tailored to the specific scenario (given the usage model of the scenario), and which probability values are automatically adjusted during system life to adapt to the evolution of the network. The idea is that, for a given assessment query (e.g. “does this TLS handshake appear to be trustworthy?”), any analyzer can issue an assessment value which directly derives from the probabilities resulting from the network evaluation. Queries will typically occur when a given host needs to evaluate the trustworthiness of a given TLS exchange. The host will run an assessment using its own Bayesian network, and will also query a random set of nodes for their assessments on the validity of the same exchange. The host will then integrate all received values and its own to get

a final assessment, and will use this assessment to decide whether to accept the TLS exchange as valid or to flag it as an intrusion. The fact that the set of analyzers is chosen randomly by the evaluating host will make it harder to manipulate the receiving assessments, provided that there are different network paths used in the communications between host and analyzers and that a majority of analyzers have not been compromised.

Apart from assessments derived from queries, some of the analysis engines (typically, the ones in network management devices) will be entitled to provide automated detection. In this way, even if an assessment has not been requested on a given TLS exchange, a network element could flag it as anomalous (e.g. if a router captures a TLS flow with a certificate belonging to a server which is known to be in a different part of the network). This will allow also to react to events not directly related to certificate forging which could enable a MITM attack, such as the isolation of a given client or server.

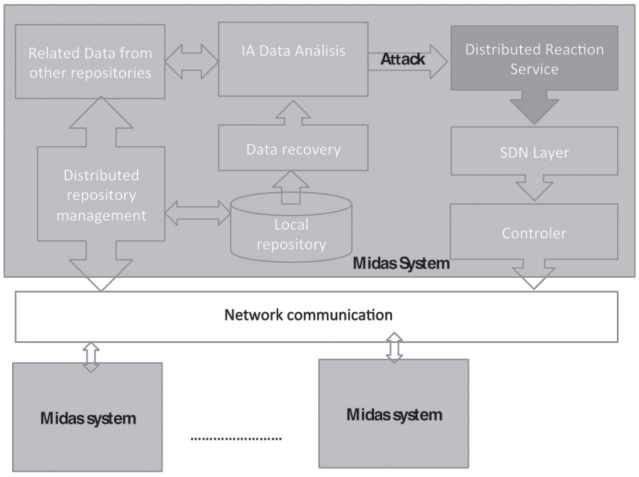
3) Reaction subsystem based on SDN

From the information obtained from the aforementioned analysis, MIDAS will be able to automatically define and put in place a restoring and reconfiguration plan of the network elements involved. This will allow, for instance, for traffic to be rerouted via an alternative path avoiding the attacking nodes, or to isolate compromised network segments. The reaction subsystem will be designed and implemented according to the novel, emerging architectural model called SDN (Software Defined Networking) that separates the control plane from the data plane in network switches and routers.

OpenFlow [23] is the first standard communications interface defined between the control and forwarding layers of an SDN architecture. It provides a singular point of control over the network flow routing decisions across the data planes of all OpenFlow-enabled network components. Taking advantage of this, security app can implement complex quarantine procedures, or malicious connection migration functions that can redirect malicious network flows in ways not easily perceived by the flow participants.

With SDN providing control over the forwarding, we can then isolate any malicious traffic to the quarantined network while all other traffic continues to operate as normal after being cleaned up. Upon detection of a potential attack, the traffic is placed in an isolated network segment that closely monitors the activity giving the attacker the perception that they’re interacting with a real system when, in reality, it’s a system that records their actions, decisions, and reactions, giving insight into their methodology.

FIGURE 3: LOGICAL ARCHITECTURE OF MIDAS SYSTEM



Although this component has not yet been implemented, similar approaches have shown its viability. In order to simplify the development and deployment process, we plan to use an approach similar to the FRESCO framework [24]. FRESCO is an OpenFlow security application development framework designed to facilitate the rapid design, and modular composition of OpenFlow-enabled detection and mitigation modules. Recently, the first security enforcement kernel for the OpenFlow controller Floodlight [25] has been released. The combination of FRESCO framework and SE-Flodlight provides a reference framework to rapidly prototype and field innovative security applications, which is makes it suitable for the kind of application that we want to develop.

4. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced MIDAS, our proposal for a distributed certificate assessment system intended to thwart advanced Man-in-the-Middle attacks. This system builds on existing network monitoring and management technologies to provide a *pinning-in-the-net* approach enabling hosts to effectively assess the validity of the certificates they encounter during TLS interactions. The system relies on the existence of a set of network probes located in different elements of the network (either hosts or switches or routers), a distributed analysis engine based on bayesian networks and a reaction subsystem which makes use of SDN technologies.

Right now we have fully implemented the network probes and developed a proof-of-concept scenario of the complete architecture. Although the system looks promising, there is still considerable work to be done to build realistic Bayesian networks specifically tailored to realistic high-sensitive network scenarios. This would result in early detection of suspicious certificates and would trigger mechanisms to defeat the attack, minimize its impact, and gather information

on the attackers. Additionally, a more detailed and thorough analysis could be performed. This would be achieved through the use of Software Defined Network (SDN) techniques, allowing a much more accurate and efficient response to man-in-the-middle attacks, and mitigating damage in highly sensitive communication networks.

5. ACKNOWLEDGMENTS

The author would like to thank Álvaro Felipe Melchor for his work in the implementation of this proposal.

REFERENCES:

[1] T. Dierks and E. Rescorla. The transport layer security (TLS) protocol version 1.2. *(5246)*, 2008. Available: http://www.ietf.org/rfc/rfc5246.txt,.

[2] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley and W. Polk. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. *(5280)*, 2008. Available: http://www.ietf.org/rfc/rfc5280.txt,.

[3] R. Oppliger. Certification authorities under attack: A plea for certificate legitimation. *Internet Computing, IEEE PP(99)*, pp. 1-1. 2013. . DOI: 10.1109/MIC.2013.5.

[4] *Comodo Report of Incident*. Available: http://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html.

[5] S. B. Roosa and S. Schultze. Trust darknet: Control and compromise in the internet's certificate authority model. *IEEE Internet Comput. 17(3)*, pp. 18-25. 2013. . DOI: http://doi.ieeecomputersociety.org/10.1109/MIC.2013.27.

[6] P. Hoffman and J. Schlyter. The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA. *(6698)*, 2012. Available: http://www.ietf.org/rfc/rfc6698.txt,.

[7] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose. DNS security introduction and requirements. *(4033)*, 2005. Available: http://www.ietf.org/rfc/rfc4033.txt,.

[8] R. Barnes. Use cases and requirements for DNS-based authentication of named entities (DANE). *(6394)*, 2011. Available: http://www.ietf.org/rfc/rfc6394.txt,.

[9] M. Marlinspike and T. Perrin. Trust assertions for certificate keys. Internet Engineering Task Force. 2013Available: http://tools.ietf.org/id/draft-perrin-tls-tack-02.txt,.

[10] C. Evans, C. Palmer and R. Sleevi. Public key pinning extension for HTTP. Internet Engineering Task Force. 27Available: http://www.ietf.org/internet-drafts/draft-ietf-websec-key-pinning-09.txt,.

[11] EFF. (feb). *The Sovereign Keys Project*. Available: https://www.eff.org/sovereign-keys/.

[12] B. Laurie, A. Langley and E. Kasper. Certificate transparency. Internet Engineering Task Force. 2013Available: http://www.ietf.org/internet-drafts/draft-laurie-rfc6962-bis-00.txt.

[13] D. Wendlandt, D. G. Andersen and A. Perrig. Perspectives: Improving SSH-style host authentication with multi-path probing. Presented at USENIX 2008 Annual Technical Conference on Annual Technical Conference. 2008, Available: http://dl.acm.org/citation.cfm?id=1404014.1404041.

[14] *The ICSI Certificate Notary*. Available: http://notary.icsi.berkeley.edu.

[15] EFF. The EFF SSL observatory. Available: https://www.eff.org/observatory 2013.

[16] *DetecTor.io*. Available: http://detector.io.

[17] M. Alicherry and A. D. Keromytis. DoubleCheck: Multi-path verification against man-in-the-middle attacks. Presented at Iscc. 2009, Available: http://dblp.uni-trier.de/db/conf/iscc/iscc2009.html#AlicherryK09.

[18] J. Braun, F. Volk, J. Buchmann and M. M\uhlh\auser. Trust views for the web PKI. *Proceedings of the 10th European Workshop on Public Key Infrastructures, Services and Application (EuroPKI 2013)* 2013.

[19] M. Abadi, A. Birrell, I. Mironov, T. Wobber and Y. Xie. Global authentication in an untrustworthy world. Presented at Proceedings of the 14th USENIX Conference on Hot Topics in Operating Systems. 2013, Available: http://dl.acm.org/citation.cfm?id=2490483.2490502.

[20] R. G. Bace. *Intrusion Detection. Ed Mc.Millan* 2000.

[21] B. Claise. Cisco systems NetFlow services export version 9. *(3954)*, 2004. Available: http://www.ietf.org/rfc/rfc3954.txt,.

[22] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Secur. 28(1–2)*, pp. 18-28. 2009. . DOI: http://dx.doi.org/10.1016/j.cose.2008.08.003.

[23] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner. OpenFlow: Enabling innovation in campus networks. *SIGCOMM Comput.Commun.Rev. 38(2)*, pp. 69-74. 2008. Available: http://doi.acm.org/10.1145/1355734.1355746. DOI: 10.1145/1355734.1355746.

[24] S. Shin, P. A. Porras, V. Yegneswaran, M. W. Fong, G. Gu and M. Tyson. FRESCO: Modular composable security services for software-defined networks. Presented at Proceedings of the 20th Annual Network & Distributed System Security Symposium. 2013, Available: http://dblp.uni-trier.de/db/conf/ndss/ndss2013.html#ShinPYFGT13.

[25] *The FloodlightProject. Floodlight*. Available: http://www.projectfloodlight.org.