International Journal of Cybersecurity Intelligence & Cybercrime

Volume 5 | Issue 2 Article 3

8-22-2022

Kerberoasting: Case Studies of an Attack on a Cryptographic Authentication Technology

Cryptographic authentication; Kerberos; Kerberoasting; Network security

IJCIC Center for Cybercrime Investigation & Cybersecurity

Demers & Lee (2022). *International Journal of Cybersecurity Intelligence and Cybercrime, 5*(2), 25-39.

Kerberoasting: Case Studies of an Attack on a Cryptographic Authentication Technology

D Demers*, Bridgewater State University, U.S.A.
Hannarae Lee, Ph.D., Bridgewater State University, U.S.A.

Keywords: Cryptographic authentication; Kerberos; Kerberoasting; Network security

Abstract:
Kerberoasting, an attack vector aimed at the Kerberos authentication protocol, can be used as part of an adversary's attack arsenal. Kerberos is a type of network authentication protocol that allows a client and server to conduct a mutual verification before providing the requested resource to the client. A successful Kerberoasting attack allows an adversary to leverage the architectural limitations of Kerberos, providing access to user password hashes that can be subject to offline cracking. A cracked user password could give a bad actor the ability to maintain persistence, move laterally, or escalate privileges in a system. Persistence or movement within a system is indispensable to a bad actor. Adversaries may use Kerberoasting to achieve this persistence or movement as part of a more effective attack. These attacks can include ransomware, stealthy removal of data from a system, or building a back door for future access. It is, therefore, vital to understand how Kerberoasting works to detect attacks and mitigate future attempts. We examine cases in which Kerberoasting has played a role in an attack or was used as a tool in an adversary's arsenal and review the outcomes. We then discuss known ways to detect and mitigate Kerberoasting attacks and analyze how this information can inform enterprise policy.

Introduction

The internet is a vital aspect of everyday life for many worldwide, including individual users, small businesses, the government, and large enterprises. Compared to its popularity, the network security of the internet is relatively insecure. Attackers can identify unencrypted passwords from vulnerable applications using password sniffing tools. In addition, many client/server applications mainly require the client to restrict its activities while the server rarely imposes any restrictions. Firewalls or authentication technologies have been implemented to accommodate better network security and minimize the existing limitations.

Among various network security approaches, this paper focuses on authentication technology, specifically a cryptographic authentication called Kerberos and its related targeted attack called Kerberoasting. Cryptographic authentication allows clients and servers to securely prove their identities before conducting business with one another to help prevent a breach. Kerberos is a specific network authentication protocol that uses cryptography to allow a client and server to verify each other over the network before providing the client access to the requested resource. Kerberoasting is the attack vector aimed at the Kerberos authentication protocol.

Kerberos authentication protocol was developed with security in mind. Even so, it is still vulnerable to attacks (Praetorian, 2022). MITRE ATT&CK is a knowledge base of Tactics, Techniques, and Procedures(TTP)

*Corresponding author
D Demers*, Acting Director of Equal Opportunity, Office of Equal Opportunity Bridgewater State University, Boyden Hall 206, 131 Summer St., Bridgewater, MA, 02325, U.S.A.
Email: d1demers@bridgew.edu

used by bad actors, which are used to create threat models for the cybersecurity community (*MITRE ATT&CK, n.d.*). MITRE ATT&CK currently lists four sub-techniques that may be used to forge or steal Kerberos tickets: Golden Ticket, Silver Ticket, Kerberoasting, and AS-REP Roasting (Praetorian, 2022). Therefore, while addressing the importance of Kerberos, this paper also focuses on Kerberoasting because there is a low investment for the attacker (i.e., it is a simple attack). If successful, Kerberoasting offers a great reward, such as lateral movement, escalated privileges, or persistence in a system. Considering the devastating damage that can be caused by a successful Kerberoasting attack, it is vital to understand not only how the attack works but how to defend against and mitigate attack attempts. While information about Kerberos and Kerberoasting is available across multiple resources, there is a lack of case studies in the literature to help gain an overall understanding of the real-life application and threat posed by this attack. The subsequent sections of this paper will cover a literature review of network security and cryptography, how Kerberos works, Kerberoasting methods, a case studies, detection and mitigation, and policy implications.

Literature Review

Network Security

Network security is an approach used to combat the security concerns of the internet. Network security relies on two basic goals: (a) prevent unauthorized individuals from accessing resources and (b) provide access to resources for individuals who are authorized (Shinder, 2001). Firewalls are one way in which some sites try to fix their network security problems. The issue with firewalls is that they are only effective if that threat is coming from outside. They are, therefore, moot when the threat is initiated from within an enterprise. Another weakness of firewalls is that they can limit how users utilize the internet. This is because increased security often means decreased access (MIT, 2021). For example, if a firewall has rules restricting websites that do not use encryption (i.e., HTTP), users' access to websites is limited to only those using encryption (i.e., HTTPS). Another example is that firewall rules can be created to block traffic on a certain Transmission Control Protocol (TCP) port. If a user is trying to access a service that utilizes a port blocked by the user's enterprise firewall, the user will receive an error message and be forbidden from accessing the resource.

A more productive way to achieve network security is by assigning access permissions, which dictate the users who have access to or are prohibited from accessing specific resources based on different circumstances. The problem with access permissions or other methods that rely on user identity is that they are only successful if the identity of the user can be verified. To mitigate this concern, authentication technologies are implemented to help confirm a user's identity (Shinder, 2001). Authentication can be used to verify the identity of an object, service, or person (Gerend, dknappettmsft, Downie, Ross, Parente, Coulter, et al., 2021). The theory behind authentication is that the provided individualized credentials verify the authentication of the user or object and grant access to a system or data (Shinder, 2001). The goal of authentication is to verify whether the object is genuine or verify that the presented service or person's credentials are authentic (Gerend, dknappettmsft, Downie, Ross, Parente, Coulter, et al., 2021).

According to Gerend and colleagues (2021), authentication is also helpful in terms of network security. In this context, the goal is to prove client-side identity to a network application or resource located on the server-side. Identity is commonly proven with a cryptographic system that uses a key known only to the user (i.e., a private key) or a shared key. During an authentication exchange, the server-side compares the received signed data with the known key. If there is a match, the authentication attempt can be validated by the server-side. The authentication process is made manageable by storing the keys in a secure central location. The recommended and default technology for storing identity information is Active Directory Domain Services, and it is required for certain default authentication protocols. Techniques for authentication include something the user knows (e.g., a password), something the user has (e.g., a token), and something the user is (e.g., biometrics). At an enterprise, several applications or resources spread across servers, either in one location or across multiple, may be accessed by services or users. This means that authentication must work for environments on other platforms and for other Windows operating systems (OS) (Gerend, dknappettmsft, Downie, Ross, Parente, Coulter, et al., 2021).

Many methods and protocols have been used to achieve the authentication exchange specific to networks. The method or protocol is usually selected based on the application and security requirements (Shinder, 2001). For the Windows OS, there are a default set of network authentication protocols used, including New Technology LAN Manager (NTLM), Transport Layer Security/Secure Sockets Layer (TLS/SSL), Digest, and Kerberos. Additionally, some protocols are joined into an authentication package. Authentication of users, computers, and services is provided by these protocols and packages, which allows those authorized to securely access the requested resources (Gerend, dknappettmsft, Downie, Ross, Parente, Coulter, et al., 2021)

Cryptography

Cryptography uses mathematical approaches to information security issues such as confidentiality; the integrity of data; and authentication of a user, server, or data origin (Menezes et al., 1997; Paar & Pelzl, 2009). Cryptography can be achieved by cryptosystems, or combinations of software and hardware which are used to convert plaintext messages to an unreadable string of characters (i.e., ciphertext) and then back to plaintext (Zwicke, 2003). This concept can be traced all the way back to ancient times, where evidence of secret writing can be found across most civilizations with developed written languages (Paar & Pelzl, 2009). By current standards, these historical ciphers, which were considered to be state of the art for their time, are significantly inadequate for achieving the level of security sought today.

The lack of security measures is cause for more significant concerns when considering the number of malicious tools available to bad actors and the fact that new tools can be easily created with just a few lines of code. For example, applications which send passwords unencrypted over a network are vulnerable to password sniffing tools used by attackers. There are also client/server applications that trust the client program to be truthful about the identity of the active user or rely on the client to restrict its own activities with no policing by the server (MIT, 2021). Increasing communication technology via the internet also requires safe communication and promoting encryption techniques, such as cryptography, digital signatures, steganography, watermarking, and other applications. Therefore, the advancement of written languages and communication technologies has driven the development of additional cryptographic techniques, especially to promote better network security. While cryptography is used to address a variety of data security issues, this paper will focus on the use of cryptography for authentication, specifically the Kerberos authentication process and its respective attack vector, Kerberoasting.

Kerberos

Authentication protocols rely on a combination of encryption and authentication methods to provide access to only those with proper authorization. A few authentication protocols exist for providing security in different scenarios. Some examples include the Secure Socket Layer (SSL), the Password Authentication Protocol (PAP), the Challenge Handshake Authentication Protocol (CHAP), and Kerberos (Pandya et al., 2015). The SSL is a common method used to give secure access to websites using a combination of symmetric encryption and public keys. The PAP, although generally insecure, is typically used to establish user authentication over a remote access control system. Shiva PAP is a variate of PAP, which encrypts the user passwords before sending it to the remote server. The CHAP uses a hashing algorithm known as MD5 to hash the password before sending it over the network to gain remote access to a system. Kerberos, a network authentication protocol, enables secure authentication for clients and servers using a secret key (Pandya et al., 2015). Even though various authentication protocols are essential to address and review to better understand network security, this paper focuses on cryptography-based authentication, Kerberos.

Kerberos was developed by the Massachusetts Institute of Technology (MIT) as a network authentication protocol to address the problems with network security. When a client and a server are connected via an insecure network, the Kerberos protocol provides a way for them to authenticate their identities to each other using strong cryptography. Once they have authenticated with each other, the client and server can also encrypt their correspondence for privacy and data integrity assurance (MIT, 2021).

Windows OS currently utilizes version five of Kerberos and extensions for public key authentication, transportation of authorization data, and delegation. A security support provider (SSP) allows for the implementation of the Kerberos authentication client, which means the Kerberos authentication client can be accessed through the Security Support Provider Interface (SSPI). The Winlogon single sign-on architecture includes initial user authentication. Kerberos employs a Key Distribution Center (KDC), which is integrated with other Windows Server security services running on the domain controller. The KDC requires a security account database, so it uses the domain's Active Directory Domain Services database (Gerend, dknappettmsft, Downie, Ross, Parente, Poggemeyer, et al., 2021).

The Kerberos authentication process can be explained in six steps, as shown in Figure 1. First, a client wishing to access a service sends a request for an authentication ticket (i.e., ticket-granting ticket [TGT]) to the authentication server. Next, the authentication server verifies the client's right to access the resources and returns an encrypted TGT and a session key. Third, the client sends the TGT, along with the requested resource, to the Ticket-Granting Service (TGS), which is usually the KDC. The TGS then provides a service-specific valid session key to the client. The client then presents the key to the service directly. In return, the service grants access to the now authorized client (Bhakhra & yogyaarora25, 2020; "Identity and Access Management," n.d.).

Halback (n.d.) provides the example of a carnival as an analogous way to think of Kerberos. When you arrive at a carnival, you have to go through the front entrance, where you receive a wrist band. This is similar to when you turn on your computer and enter your credentials to access your corporate network. During this process, your computer requests the TGS (a wrist band) from the domain controller to enter the realm (carnival). Once inside the carnival, you must go to a ticket booth and purchase tickets in order to ride the rides. The rides are equivalent to accessing a file share server containing your work files and the tickets are equivalent to the service-specific valid session key and ticket. Once you receive tickets (ticket and session key), you wait in line for a specific ride (file server). The ride attendant confirms that you are tall enough and of age to ride, takes your ticket, and grants you access onto the ride. This is akin to the client presenting the session key and ticket to the file server, which would provide access to the files, providing the client has permission. Your ticket would only allow you access to the ride (server) that one time. If you wanted to ride again (access the server again), or ride another ride (access a different server), you would have to revisit the ticket booth and start the process over again (Halbach, n.d.).

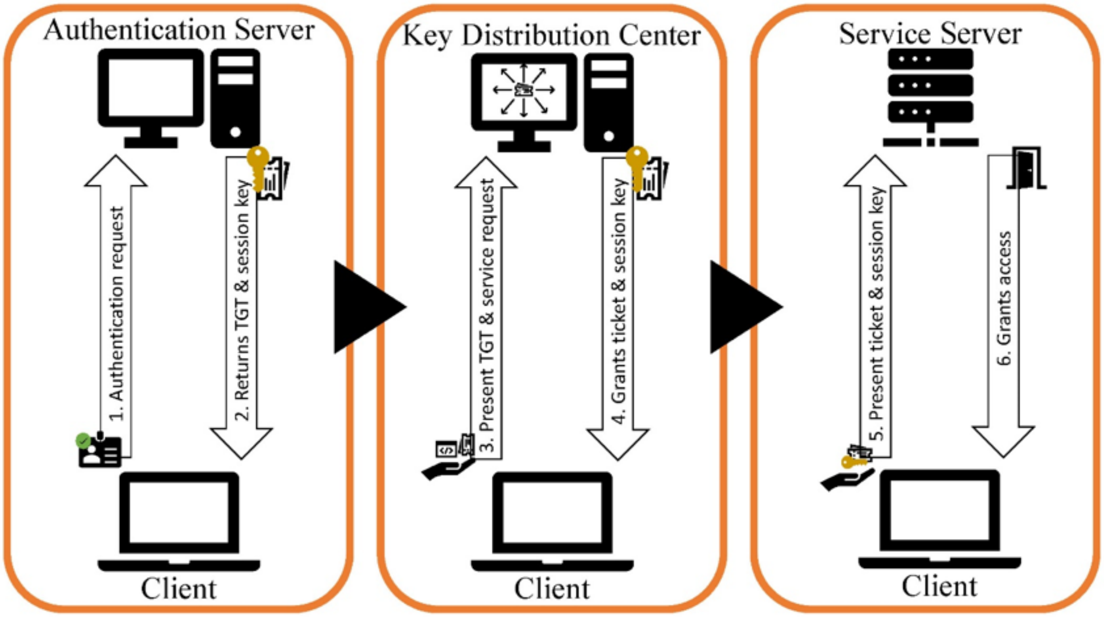
Authentication Server
1. Authentication request
2. Returns TGT & session key
Client
Key Distribution Center
3. Present TGT & service request
4. Grants ticket & session key
Client
Service Server
5. Present ticket & session key
6. Grants access
Client

Figure 1. Kerberos authentication process

The benefits of using Kerberos for domain-based authentication are delegated authentication, single sign on, interoperability, streamlined authentication to servers, and mutual authentication. Delegated authentication is achieved because Kerberos authentication allows for service to act in the place of its client . Single sign on allows a user or service to maintain access to resources without repeated credential requests. Kerberos maintains credentials after the initial Winlogon domain sign on. The Kerberos protocol provides a foundation for working with other networks using the same protocol, making it interoperable. Kerberos streamlines authentication to servers by alleviating the need for pass-through authentication and replaces it with renewable session tickets. Kerberos does not assume that servers are genuine. Therefore, it allows for mutual authentication, meaning that the server can verify the client, but the client can also verify the server (Gerend, dknappettmsft, Downie, Ross, Parente, Poggemeyer, et al., 2021).

Some additional advantages to using Kerberos as a network authentication protocol include the difficuly of reusing a ticket due to their quick expiration and strong authentication needs, the constant encryption of passwords, and the efficiency granted by sharing secret keys. There are, however, a couple of disadvantages to note. One limitation is that Kerberos is vulnerable to weak or repeated passwords. The other is that authentication is limited to servers and clients (Bhakhra & yogyaarora25, 2020; Tawde, n.d.).

Kerberoasting

Kerberoasting was discovered by Tim Medin, who presented on the attack for the first time in 2014 at DerbyCon (Kotlaba et al., 2020, 2021; Medin, 2014; *Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021). The goal of Kerberoasting is to retrieve credentials of Active Directory service accounts without permission or escalation of privileges (Kotlaba et al., 2020; *Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021), and it is one of the most common attacks suffered by domain controllers (Ghosal, n.d.). It is pervasive in nature and lends itself to bad actors with any level of experience because the technique does not need to be carried out by an administrator; instead, it can be implemented by any user on a domain (*Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021). The only prerequisite to Kerberoasting is that an attacker must already have access to a compromised system, but a complete compromise of the Windows domain is not required (Medin, 2014; Motero et al., 2021; *Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021).

As previously mentioned, the TGS provides a service ticket to a client based on the specific resource to which the client requested access. These TGS tickets are then encrypted using RC4_HMAC_MD5 (i.e., the New Technology LAN Manager (NTLM) hash of the service account belonging to the resource requested by the client). Service Principal Names (SPNs) are used in Windows to identify which service accounts are used to encrypt TGS tickets. SPNs can be linked to either domain user or host-based accounts. Host-based accounts have a randomly generated 128-character long password which is changed every 30 days (Metcalf, 2017a; Motero et al., 2021; *Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021; Schneider, 2018). The significant length of the password and its short lifespan make it practically impossible to guess, even with a modern kit of password cracking tools and powerful hardware (Motero et al., 2021; *Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021). Kerberoasting instead relies upon the fallibility that comes from passwords generated by humans for domain user accounts. These passwords are often weak, seldom updated, and thus, easily cracked (Motero et al., 2021; *Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021; Schneider, 2018).

This is where the Kerberos authentication protocol becomes vulnerable. The protocol allows a domain user to request a TGS ticket from a domain controller for any service on the network with a registered SPN. The domain controller fails to check whether the user making the request has the authorization to access the resource. The service is responsible for credential verification, which creates an opportunity for an offline attack (*Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021; Schneider, 2018). The NTLM hash encryption of the ticket allows for the ticket to be subject to brute force attacks offline in an attempt to expose credentials stored in plaintext (Praetorian, 2022; Schneider, 2018). Simply stated, in a Kerberoasting attack, a TGS service ticket can be requested by a valid domain account for any service, which allows for the opportunity to crack passwords offline using the ticket (Schneider, 2018).

Thinking about the carnival analogy, unlike ride tickets which are printed with basic ticket information, the tickets in the Kerberos process have an abundance of information printed on them. This information includes the domain name of the service account, and the ticket is encrypted with the client's account password hash. While this password hash is necessary for the Kerberos process to finish, it is also the focus of bad actors for use in offline password cracking attempts (Halbach, n.d.). If successfully cracked, this information can allow a hacker to maintain persistence, escalate privilege, or move laterally in the system (Medin, 2014; Praetorian, 2022).

Kerberoasting attacks are prevalent because they require basic skills to implement and their very nature shields adversaries from detection. A successful Kerberoasting attack allows an adversary to navigate a system masked as an approved user. It is atypical for a cybersecurity detection tool to analyze behavior of approved users. Acting as an approved user makes it unlikely that these tools would detect adversary activity. These attacks also do not require the use of malware, making them immune to defensive technologies such as antivirus software. The fact that adversaries can conduct the password cracking offline means that they do not produce any suspicious network traffic or transmit data packets. This means the activity is neither logged, nor does it trigger an alert (Shastri, 2022). To help better understand Kerberoasting, the following section provides several case studies.

Kerberoasting Case Studies

In recent years, the use of Kerberoasting has been documented as one of the TTPs in reports analyzing the attack modus operandi of particular eCrime groups or their attacks. Kerberoasting is not the sole focus in the following case studies, however it is a critical aspect of each case. The impact of having Kerberoasting in each adversary's arsenal was significant enough in the following cases to have all been included in MITRE ATT&CK's procedure examples for Kerberoasting (Praetorian, 2022).

Operation Wocao

The first case is from Fox-IT, a cyber security group providing different incident response services to their clients, such as crisis management, technical investigations, and remediation. In 2019, they published a report of a profile they had built over two years about a Chinese-based hacking group whom they refer to as Operation Wocao. They found that little was known about Wocao. Thus, Fox-IT sought collaboration across the public sector to help determine, with medium confidence, that the techniques and tools they were encountering were being used by Wocao, also known as APT20 within the industry. Fox-IT made the following two determinations. First, in order to execute code on the system being attacked, Wocao used PowerSploit's Invoke Mimikatz module to try and extract Kerberos tickets from the system's memory (van Dantzig & Schamper, 2019).

Second, Wocao used PowerSploit's Invoke-Kerberoast module to try to solicit credentials. Invoke-Kerberoast allows Wocao to request encrypted service tickets by brute forcing Windows service account passwords (Praetorian, 2022; van Dantzig & Schamper, 2019). Both actions aimed to maintain persistence in the system, achieve lateral movement, and eventually escalate privileges. Wocao then used the obtained privileges from the successful Keberoasting-based attacks to exfiltrate data of interest in the system before covering their footprints and cutting access (van Dantzig & Schamper, 2019).

Carbon Spider

The second case is from CrowdStrike, which provides a platform aimed at stopping data breaches to its customers. In 2021, CrowdStrike published a two-part report on Carbon Spider, also known as FIN7, and their use of ransomware. In 2020, Carbon Spider shifted their modus operandi from campaigns focused on companies with point-of-sale devices to using another adversary's ransomware to launch large attacks with the simple goal of infecting as many victims across all sectors as possible. This type of cyberattack is referred to as Big Game Hunting (BGH). In August 2020, Carbon Spider began using their ransomware, called Darkside. Then, in November 2020, Carbon Spider launched a Darkside ransomware-as-a-service scheme, allowing other bad actors to employ the ransomware while returning a percentage of the ransom obtained back to Carbon Spider. CrowdStrike was able to identify several Darkside campaigns that all utilized similar tools to achieve initial access (Loui & Reynolds, 2021). After gaining this initial access, Carbon Spider's next step was to harvest credentials and enable lateral movement (Loui & Reynolds, 2021; Praetorian, 2022). In addition to CrackMapExec, Mimikatz, PowerSploit, and SessionGopher, one of the tools consistently used for this task was Kerberoasting. Once in possession of valid credentials, Carbon Spider uses them to move laterally through a compromised system, encrypting files and exfiltrating data to maliciously leak along the way (Loui & Reynolds, 2021).

Nobelium

The third case is based on the Microsoft Threat Intelligence Center (MSTIC) analysis of the attack against SolarWinds by an adversary called Nobelium by MSTIC but known as APT29 or CozyBear within the industry. This attack began in September 2019 and continued periodically until December 2020 (Microsoft 365 Defender Research Team et al., 2021). During this phishing and spear-phishing attack, the adversary began stealing credentials and conducting lateral movements to search for valuable assets through a backdoor they were able to build in the target system. Nobelium was able to maintain long-term persistence and extract vital data from SolarWinds. In addition, they were also able to use SolarWinds' supply chain to access high-profile victims (Cadieux, 2021). The depth of details unearthed during the investigation earned this attack, called Solarigate, the title of "one of the most sophisticated and protracted intrusion attacks of the decade" (Microsoft 365 Defender Research Team et al., 2021, para. 1).

Microsoft 365 Defender's telemetry revealed that Nobelium used a vast array of TTPs during Solarigate. These TTPs were carefully curated and implemented for maximum camouflage in SolarWinds' environment. MSTIC highlighted 16 TTPs used in Solarigate by Nobelium that they found to be the most interesting. Included on this list was Kerberoasting (Microsoft 365 Defender Research Team et al., 2021). Nobelium utilized Kerberoasting to obtain TGS tickets for Active Directory SPNs (Microsoft 365 Defender Research Team et al., 2021; Praetorian, 2022). The actual loss and damage from the Nobelium are not specified, but since they targeted diplomats worldwide, at least one successful attempt may cause unmeasurably devastating consequences.

Wizard Spider

In 2020, Ryuk, ransomware operated by eCrime group Wizard Spider, made a dramatic re-entry into the spotlight when it was used in attacks against Universal Health Service hospitals, managed service providers, and a furniture manufacturer called Steelcase (Hanel, 2019). This campaign of attacks, specifically those targeted at the healthcare sector, was cause for such concern that the Cybersecurity and Infrastructure Security Agency (CISA) issued an alert. This alert identified the Ryuk campaign as an imminent threat to the healthcare and public health sector (CISA et al., 2020). The Digital Forensics and Incident Response (DFIR) Report published full reports on a few of the attacks. The DFIR Report did not specify which attacks they analyzed. However, they provide ample details related to the timeline and TTPs organized according to the MITRE ATT&CK framework (*Ryuk's Return, 2020; Ryuk Speed Run, 2 Hours to Ransom*, 2020). Therefore, the following two case studies focused on the Ryuk attacks in which Kerberoasting was used. For clarity purposes, the first case discussed is referred to as Email Attack and the second is called Download Attack.

The email attack. The attack took 29 hours to execute. During that time, the adversary successfully employed their ransomware and demanded over 600 Bitcoins, amounting to over six million US dollars at the time. As soon as the payload was executed, the Bazar/Kegtap malware penetrated several processes with the goal of running discovery utilizing Windows built-in utilities (e.g., nltest, netgroup) and a third-party tool called AdFind. After the first wave of discovery on day one, the malware sank to the background only to reemerge the next day to continue discovery. Rubeus was added on the second day so the adversary could attempt Kerberoast to retrieve Advanced Encryption Standard (AES) hashes (Praetorian, 2022; *Ryuk's Return*, 2020). Shortly after, the adversary began lateral movements. They eventually exfiltrated the AdFind and Rubeus outputs before running additional commands and executing the ransom. As noted in the report, if the first day of recon went unnoticed by a defender, that would leave the defender just over three hours to launch a defense before the ransom was set (*Ryuk's Return*, 2020).

The download attack. The attack took only two hours, 27 hours less than the Email Attack. The Download Attack was initiated and completed in less time than the defenders would have had to react on day two of the Email Attack. The Download Attack was initiated through a phishing email with malicious links to Google Drive that, when opened, downloaded Bazar malware. Domain discovery began within five minutes, followed by the execution of Cobalt Strike. Once again, AdFind and Rubeus were used to harvest credentials for exfiltration using Kerberoasting techniques. The attack continued until the adversary was able to encrypt the entire domain, at which point they held it for ransom (*Ryuk Speed Run*, 2 Hours to Ransom, 2020).

From the case studies discussed in this paper, it is apparent that bad actors use Kerberoasting attack methods to maintain persistence in the system or escalate privileges in order to achieve various end goals, including data exfiltration, ransomware, and phishing attacks. These case studies all assert that Kerberoasting is a vital part of the attack plan and provides a path to more extensive and more successful attacks. The next sections address ways to detect and mitigate against Kerberoasting attacks.

Detecting Kerberoasting Attacks

According to MITRE ATT&CK, logging Kerberos TGS service ticket requests using Audit Kerberos Service Ticket Operations is one way to detect a Kerberoasting attack. When deploying this method, particular attention should be paid to any suspicious activity patterns, such as accounts making several requests in a short period of time, especially if the account also requested RC4 encryption (Praetorian, 2022). The logs of these events should be filtered for Kerberos TGS service tickets with RC4 hash encryption (Type 0x17) and sent into a Security Information and Event Management (SIEM) tool. Additional filters can include excluding requests from service accounts, turning on Audit Success, and excluding requests for service names, including a $ (Metcalf, 2017b).

Another option is to deploy a honeypot account with a fake but unique SPN. A honeypot is a fake account intentionally designed to be alluring to an adversary that can be used to detect suspicious activity. Any request for the honeypot account's fake SPN will not be valid, indicating that it is malicious. It is beneficial to make this honeypot account more enticing. Two ways to do that are to set the AdminCount attribute to one, signifying that the account may have elevated rights, and by adding the account to fake groups to create the impression that it is granting additional elevated rights. Once the account is live, any attempt to access it should be investigated. As the account is fake and is not linked to any real application, there is no reason to request a TGS service ticket. Therefore, it is probable that anyone trying to request such a ticket is deploying a Kerberoast attack on the fake account (Metcalf, 2017b).

A third option is to employ the use of a third-party monitoring technology such as those offered by Qomplx or Awake Security Platform. Qomplx technology watches for Kerberoasting attack signs, such as Event ID 4769. They offer a Privilege Assurance tool, which operates on the principle of least privilege, minimizing the chance of service accounts having too many permissions. Qomplx also monitors Windows Event Logs for suspicious activity and compares them to other logs in order to establish behavioral indicators of attempted Kerberoasting activity (*Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021). Awake Security Platform utilizes an abstraction layer that allows them to analyze Lightweight Directory Access Protocol (LDAP) search Request messages and Kerberos authentication protocol data to identify suspicious activity. When the Awake Security Platform identifies a suspicious activity, it uses a graphical representation to show that there was a Kerberoasting attempt (Ghosal, n.d.). Enterprises seeking the services of a third-party platform should research options, request demos, and make a determination based on their enterprise needs.

Mitigation of Kerberoasting

Fortunately, MITRE ATT&CK identifies ways to mitigate against a Kerberoasting attack. The first is encrypting sensitive information (Praetorian, 2022). Kerberoasting is, in part, successful because of the weak RC4 encryption. AES is now offered as a Kerberos encryption option, which is beneficial because of the stronger hashing it employs (Metcalf, 2015, 2017a). Therefore, AES Kerberos encryption, or another encryption standard similar in strength, should be used in place of RC4 encryption wherever possible (Praetorian, 2022).

The second is implementing strong password policies. Password requirements should dictate that passwords are changed regularly and at least 25 characters long, preferably 30 or more, which increases the difficulty of cracking (Metcalf, 2015, 2017a; Praetorian, 2022). Practicing password rotation puts time constraints on attackers attempting to decrypt long, complex passwords (*Qomplx Knowledge: Kerberoasting Attacks Explained*, 2021). Managed Service Accounts and Group Managed Service Accounts can help ensure that passwords meet the length and strength requirements and change regularly. An additional solution is employing a third-party technology to securely store passwords in a password management software (i.e., a vault) and using that vault to sign into other accounts (Metcalf, 2015, 2017a; Praetorian, 2022). It is crucial to evaluate any third-party password management tool considering that the associated service account would likely require Domain Administrator rights (Metcalf, 2017a).

The third mitigation recommendation from MITRE ATT&CK is privileged account management. It is not uncommon for service accounts to be over-privileged, having full administrator rights to Active Directory when not required (Metcalf, 2015, 2017a). Service accounts should follow the principle of least privilege and the minimal access necessary to perform their function. This includes privileged group membership such as Domain Administrators (Praetorian, 2022).

Policy Implications

Based on the cases discussed in this paper, it is clear that Kerberoasting is used as a steppingstone to help achieve a more significant and more detrimental attack. Oftentimes, the goal of the overall attack is to encrypt a system and hold it for ransom, as seen with Carbon Spider, Nobelium, and Wizard Spider. When facing ransomware, there are generally only two outcomes: (a) the enterprise pays a significant ransom to regain access to their files, or (b) the enterprise does not pay and loses access to their data forever. However, as seen in the Fox-IT case, ransomware is not the only attack in which Kerberoasting can play a role. Kerberoasting can also be used to help an attacker quietly copy data from a domain before evacuating. In turn, this data can be used against an enterprise or sold on data leak sites. Either option will likely cause the enterprise money and resources. Considering that Kerberoasting is part of an attack arsenal and not a substantial threat on its own, the question becomes: is it worth defending against Kerberoasting attacks? The short answer is YES.

A common term used in the world of cybersecurity is Defense in Depth. The Defense in Depth approach is frequently referenced by the National Institute of Standards and Technology (NIST) in several of their publications (Computer Security Resource Center, n.d.). The Center for Internet Security (CIS) explains that a goal of a Defense in Depth approach is redundancy within network security with the aim of preventing any single point of failure. One of the ways it achieves this is by increasing the time and complexity needed to compromise a network successfully. In turn, this not only depletes the resources of the bad actors but also makes it more likely that an active attack can be identified and addressed before its completion (Center for Internet Security, n.d.).

This is not a new approach to securing valuable assets (Center for Internet Security, n.d.). Defense in Depth approach can be seen in countless security measures today, such as homes surrounded by fences, and equipped with locked doors; video-monitoring doorbells; a security system; or banks with a security guard, protective glass in front of the tellers, time-controlled safes, and panic buttons. We can even look back in history and see this method used with the outer walls, moats, towers, lookout points, and drawbridges in castles (Cyber Edu, n.d.).

Practically, the architecture of a Defense in Depth approach can be broken down into physical controls, technical controls, and administrative controls. Physical controls prevent physical access to information and communications technology (ICT) systems. Technical controls protect network systems, hardware, and software.

Administrative controls are comprised of policies and procedures for employees. Additional security layers that may help protect different elements of a network include access measures, workstation defenses, data protection, perimeter defenses, and monitoring and prevention (*Defense-in-Depth*, n.d.).

With a Defense in Depth approach in mind, it becomes easier to see how preparing defenses against Kerberoasting and other small tools used in more significant attacks can help on a bigger scale. By strengthening the Kerberos network authentication protocol, an enterprise can halt an attack by removing the adversary's ability to harvest credentials and achieve lateral or upward movement, or persistence. Suppose a stronger Kerberos protocol does not circumvent an attack. In that case, it may give the defense team time to prepare and react to an attack by delaying the speed with which an adversary is able to navigate through their domain. As seen with the Carbon Spider and Wizard Spider ransomware attacks, time is valuable. If an adversary is able to compromise an enterprise's system from start to finish within a matter of a few hours, that limits the defense team's ability to protect their assets. Detection of a Kerberoasting attempt is comparable to an early warning system. As indicated in the case studies, Kerberoasting is typically deployed earlier in the attack considering it is usually used to maintain persistence, move laterally, or escalate privilege, which is often needed to continue into the next stages of an attack (Medin, 2014; Praetorian, 2022). Coupled with the fact that Kerberoasting is a good indicator that an adversary has a larger goal, and therefore attack, in mind, both detection and mitigation of Kerberoasting provide an enterprise's cybersecurity team with a couple of solid layers of defense.

ICT has experienced rapid growth and development over the last few decades. As technology advances, so do attack vectors created and used by bad actors. Currently, a limited number of post-attack reports are publicly shared by enterprises after they have suffered an incident. NIST's Computer Security Incident Handling Guide summarizes the incident response cycle in four steps: (1) preparation; (2) detection and analysis; (3) containment, eradication, and recovery and; (4) post-incident activity. They say that one of the essential parts of incident response is that final step, specifically learning and improving. It can help an enterprise enhance security measures and adapt to evolving technologies and threats. NIST notes that information sharing between enterprises is the most important element of incident response coordination (Cichonski et al., 2012).

Enterprises often face the same threats, especially within the same industry. Sharing information can be beneficial across the board. If a cybersecurity expert sees suspicious activity in their environment, they should be able to share that with their industry and see what other cybersecurity experts from other enterprises may be doing in response. If more enterprises were willing to share their reports, it would help to shed light on the TTPs used across adversary groups. Analyzing each of these attacks next to one another may assist enterprises in increasing their detection and mitigation defenses, thereby limiting the impact caused by these attacks.

Conclusion

Network security is crucial because, while the internet can be a treasure trove of information and resources, it ultimately falls short in terms of safety. One way of ensuring network security is through authentication. There are many methods and protocols for authentication exchange depending on the application and security requirements, such as Kerberos, which is designed to provide a way for a client and a server to authenticate each other's identities using cryptography over an insecure network. While Kerberos has several benefits, it may be vulnerable to Kerberoasting attacks if a bad actor already has access to a compromised system. Although Kerberoasting cannot be performed against a host-based account due to the strong password policies, it is useful against domain user accounts because of weak passwords protected by equally weak encryption.

Based on the review of eCrime groups and the common TTPs, the most robust protection against Kerberoasting is a Defense in Depth approach. This approach should include both detection and mitigating factors. Tracking and reviewing filtered logs is the best way to detect Kerberoasting attempts. Another way to detect Kerberoasting attempts is to create a honeypot account. An attempt to access the honeypot is a good indication of a Kerberoasting attempt considering that there is no reason to access the account. Lastly, third-party applications can also provide monitoring services. Mitigation of Kerberoasting can be achieved in three ways: employing more robust Kerberos encryption (e.g., AES), implementing strong password policies, and managing service accounts based on the principle of least privilege.

In conclusion, while Kerberoasting poses a threat to the Kerberos authentication process, there are a few simple implementations that can mitigate these attacks, thereby rendering the authentication process much more secure. Implementation of these measures, coupled with other defensive measures geared toward other common attack vectors and a sharing of information between enterprises, can, at most, help an enterprise avoid a breach altogether; and, at least, give the defense team time to react to an attack as it is happening and mitigate the outcome.

References

Bhakhra, S., & yogyaarora25. (2020, September 17). *Kerberos. GeeksforGeeks.* https://www.geeksforgeeks.org/kerberos/

Cadieux, L. (2021, February 24). *SolarWinds & Solarigate: What happened, why it matters & what happens next.* The Devolutions Blog. https://blog.devolutions.net/2021/02/solarwinds-solorigate-what-happened-why-it-matters-what-happens-next/

Center for Internet Security. (n.d.). *Election security spotlight - Defense in Depth (DID).* https://www.cisecurity.org/spotlight/cybersecurity-spotlight-defense-in-depth-did/

Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). Computer Security Incident Handling Guide. *NIST Special Publication 800-61, Revision 2.* https://doi.org/10.6028/NIST.SP.800-61r2

Computer Security Resource Center. (n.d.). *defense-in-depth.* NIST: Information Technology Laboratory. https://csrc.nist.gov/glossary/term/defense_in_depth

Cybersecurity and Infrastructure Security Agency (CISA), Federal Bureau of Investigation (FBI), & Department of Health and Human Services (HHS). (2020, November 2). *Alert (AA20-302A): Ransomware activity targeting the healthcare and public health sector.* CISA: National Cyber Awareness System. https://www.cisa.gov/uscert/ncas/alerts/aa20-302a

Defense-in-Depth. (n.d.). Imperva. https://www.imperva.com/learn/application-security/defense-in-depth/

Gerend, J., dknappettmsft, Downie, K., Ross, E., Parente, J., Coulter, D., Krejcha, J., Poggemeyer, L., Hall, J., & JanKeller1. (2021, July 29). Windows authentication overview. Microsoft. https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/windows-authentication-overview

Gerend, J., dknappettmsft, Downie, K., Ross, E., Parente, J., Poggemeyer, L., & Hall, J. (2021, July 29). *Kerberos authentication overview.* Microsoft. https://docs.microsoft.com/en-us/windows-server/security/kerberos/kerberos-authentication-overview

Ghosal, S. (n.d.). *Kerberoasting: Threat hunting for Active Directory attacks.* Awake Security. https://awakesecurity.com/blog/kerberoasting-threat-hunting-for-active-directory-attacks/

Halbach, B. (n.d.). *A guide to Kerberoasting*. RedTeam Security. https://www.redteamsecure.com/research/a-guide-to-kerberoasting

Hanel, A. (2019, January 10). *Big game hunting with Ryuk: Another lucrative targeted ransomware*. Crowd Strike. https://www.crowdstrike.com/blog/big-game-hunting-with-ryuk-another-lucrative-targeted-ransomware/

Identity and access management. (n.d.). In C*ompTIA SY0-601.AE1*. uCertify. https://www.ucertify.com/?func=ebook&chapter_no=9#top.

Kotlaba, L., Buchovecká, S., & Lórencz, R. (2021). Active Directory Kerberoasting attack: Detection using machine learning techniques. *ICISSP 2021 - Proceedings of the 7th International Conference on Information Systems Security and Privacy*, 376–383. https://doi.org/10.5220/0010202803760383

Kotlaba, L., Buchovecká, S., & Lórencz, R. (2020). Active Directory Kerberoasting attack: Monitoring and detection techniques. *ICISSP 2020 - Proceedings of the 6th International Conference on Information Systems Security and Privacy*, 432–439. https://doi.org/10.5220/0008955004320439

Loui, E., & Reynolds, J. (2021, August 31). *Carbon Spider embraces big game hunting, part 1.* CrowdStrike. https://www.crowdstrike.com/blog/carbon-spider-embraces-big-game-hunting-part-1/

Medin, T. (2014). Attacking kerberos: Kicking the guard dog of Hades. *DerbyCon*. https://doi.org/https://www.redsiege.com/wp-content/uploads/2020/08/Kerberoastv4.pdf

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1997). *Handbook of applied cryptography*. CRC Press. https://doi.org/10.1201/9780429466335

Metcalf, S. (2015, December 31). C*racking Kerberos TGS tickets using Kerberoast: Exploiting Kerberos to compromise the Active Directory Domain*. Active Directory Security. https://adsecurity.org/?p=2293

Metcalf, S. (2017a, February 5). Detecting Kerberoasting activity. Active Directory Security. https://adsecurity.org/?p=3458

Metcalf, S. (2017b, February 8). *Detecting Kerberoasting activity part 2: Creating a Kerberoast service account.* Activity Directory Security. https://adsecurity.org/?p=3513

Microsoft 365 Defender Research Team, Microsoft Threat Intelligence Center (MSTIC), & Microsoft Cyber Defense Operations Center (CDOC). (2021, January 20). *Deep dive into the Soligate second-stage activation: From SUNBURST to TEARDROP and Raindrop*. Microsoft Security. https://www.microsoft.com/security/blog/2021/01/20/deep-dive-into-the-solorigate-second-stage-activation-from-sunburst-to-teardrop-and-raindrop/

MIT. (2021, July 25). *Kerberos: The network authentication protocol.* MIT.Edu. http://web.mit.edu/KERBEROS/

MITRE ATT&CK. (n.d.). https://attack.mitre.org/

Motero, C. D., Higuera, J. R. B., Higuera, J. B., Montalvo, J. A. S., & Gómez, N. G. (2021). On attacking Kerberos authentication protocol in Windows Active Directory services: A practical survey. *IEEE Access, 9*, 109289–109319. https://doi.org/10.1109/ACCESS.2021.3101446

Paar, C., & Pelzl, J. (2009). *Understanding cryptography: A textbook for students and practitioners*. Springer.

Pandya, D., Narayan, K. R., Thakkar, S., Madhekar, T., & Thakare, B. S. (2015). An overview of various authentication methods and protocols. *International Journal of Computer Applications, 131*(9). https://doi.org/10.5120/ijca2015907389

Praetorian. (2022, March 8). *Steal or forge Kerberos tickets: Kerberoasting*. MITRE ATT&CK. https://attack.mitre.org/techniques/T1558/003/

Qomplx Knowledge: Kerberoasting attacks explained. (2021). Qomplx. https://www.qomplx.com/qomplx-knowledge-kerberoasting-attacks-explained/

Ryuk's return. (2020, October 8). The DFIR Report. https://thedfirreport.com/2020/10/08/ryuks-return/

Ryuk speed run, 2 hours to ransom. (2020, November 5). The DFIR Report. https://thedfirreport.com/2020/11/05/ryuk-speed-run-2-hours-to-ransom/

Schneider, M. (2018, October 11). *Kerberoasting: Stealing service account credentials.* Scip AG. https://www.scip.ch/en/?labs.20181011

Shastri, V. (2022). *Kerberoasting attacks.* CrowdStrike. https://www.crowdstrike.com/cybersecurity-101/kerberoasting/

Shinder, D. (2001, August 28). *Understanding and selecting authentication methods*. TechRepublic. https://www.techrepublic.com/article/understanding-and-selecting-authentication-methods/

Tawde, S. (n.d.). *Kerberos*. EDUCBA. https://www.educba.com/kerberos/

van Dantzig, M., & Schamper, E. (2019, December 19). O*peration Wocao: Shining a light on one of China's hidden hacking groups.* Fox-IT. https://www.fox-it.com/media/kadlze5c/201912_report_operation_wocao.pdf

Zwicke, A. (2003). An introduction to modern cryptosystems. *Global Information Assurance Certification (GIAC)*, 1–11.