Global Information Assurance Certification Paper

Linux/Unix v. Ransomware: How Ransomware Attacks Inform the Defense of Linux & Unix Systems

GIAC (GCUX) Gold Certification

Author: David Kennel, dakennel@gmail.com

Advisor: Richard Carbone

Accepted: March 9, 2017

Abstract

Ransomware is serious business for attackers who are now raking in record amounts from businesses and consumers. Increasing threefold in 2016, these attacks pose a serious threat to all types of organizations, many of which have been caught without effective defenses and have been forced to pay the ransom demands. Some victims have paid the ransom but did not receive a valid decryption mechanism resulting in lost funds and lost data.

Many Unix/Linux system administrators may be tempted to dismiss ransomware as another Windows vulnerability; this is not the case. Researchers have recently discovered ransomware variants with cross-platform capabilities and there have been several waves of attacks against poorly secured big data systems like MongoDB. Because Linux servers house tremendous quantities of data and are key to business operations, Linux and Unix systems administrators should expect that they would increasingly be a target for this type of attack. The following discussion will examine the patterns of ransomware attack behavior from industry analysis of existing ransomware samples and existing research conducted on ransomware attacks. From these patterns, and expected growth, Linux/Unix capabilities will be explained to understand how defenses, standards and policies for Unix/Linux systems should be adjusted to help defend systems against ransomware attacks.

1. Introduction

Crypto-ransomware has grown in sophistication and breadth over the last three years. It has seriously affected many organizations including the operations of police departments and hospitals (Krebs, 2016). The attacks are growing rapidly in volume, according to a Kaspersky Lab report, with attacks against corporations rising from "27,000 between 2014 and 2015 to 158,000 between 2015 and 2016" (Kaspersky Lab, 2016). What was once easily dismissed as a problem strictly affecting the Microsoft Windows operating system, ransomware is now a cross-platform problem that affects Linux and UNIX systems, including Apple's MacOS. Linux systems have been the direct victims of ransomware (Dr. Web, 2015) and have been used indirectly as a pivot point in ransomware attacks against other systems (Hitchcock/Alert Logic Security Research, 2016).

In the tradition of offense informing defense, there are defensive strategies that can be learned from examining the patterns and techniques of existing ransomware attacks. Informed predictions on the future development of this type of attack can also be extrapolated based on the evolution of ransomware attacks. With these patterns and predictions in hand, it is possible then to look at Linux/Unix technical capabilities and make recommendations for defensive techniques and changes to system operations policies that would help defend Linux and Unix operating systems from ransomware attacks.

This paper is based on current research into ransomware and uses large studies of ransomware and detailed reports on the specific functionality of ransomware samples as source material. It also includes information from news reports on recent ransomware activity.

2. Ransomware: A Brief Introduction and History

Ransomware is, essentially, a denial of service (DOS) attack against data. Striking at the availability component of the confidentiality, integrity and availability (CIA) triad central to information security, ransomware attacks deny access to data by encrypting it

and then extorting money for access to the keys necessary to decrypt the data. In modern ransomware samples, criminals usually demand payment in Bitcoin, but pre-paid online mechanisms such as Moneypack and Ukash, and premium SMS messages have also been used (Kharraz, Robertson, Balzarotti, Bilge & Kirda, 2015).

Crypto-ransomware has been an evolving threat to computing systems since its surprisingly early origins. The first ransomware sample may be the AIDS Trojan from 1989. This DOS malware distributed on floppy disks via the mail invoiced the user for $189 or $378 US then encrypted the names of files on the system to ensure payment (Solomon, Nielson & Meldrum, n.d.). While this first example of such an attack was not very successful, criminals would come back to the concept of extortion via software in 2005 with the introduction of the next crypto-ransomware sample, trojan.Gpcoder (Savage, Coogan & Lau, 2015). Malware authors tried several different evolutions of crypto-ransomware between 2005 and 2008 after which the criminals switched tactics (Savage, Coogan & Lau, 2015). Fake anti-virus software and alerts were tried, then "locker" software which locked the user out of their PC (Savage, Coogan, & Lau, 2015). The modern outbreak of crypto-ransomware started in 2013 with the outbreak of Cryptolocker (Kelion, 2013).

For the Linux/Unix community crypto-ransomware is a relatively new phenomenon. The first known ransomware sample to attack Linux hosts is Linux.Encoder.1 (a.k.a., Linux/Ransm-C, ELF/Filecoder.A, Trojan-Ransom.Linux.Cryptor.a), first reported in the fall of 2015, which attacked Linux webservers (Dr. Web, 2015). Following quickly thereafter was the KeRanger malware affecting MacOS systems (Xiao & Chen, 2016). Analysis of this malcode would later identify it as a port of Linux.Encoder to the Mac platform (Bitdefender Labs, 2016). The most recent and probably the most effective attacks against Linux/Unix systems have been the targeting of insecure big data systems like MongoDB, Hadoop, CouchDB and ElasticSearch (Cimpanu, 2017). This indicates recognition by ransomware gangs of the importance of Unix/Linux systems in many companies computing infrastructure.

3. Ransomware Attack Patterns

The attack process for most crypto-ransomware samples are similar and contains the following steps:

- Select the target(s)
- Execute malcode on the target system
- Encrypt data
- Exchange key material with C&C server
- Make ransom demand and provide link to payment method
- Decrypt data

Each of these steps is discussed in more detail below.

Most ransomware is targeted for broad distribution and seeks to maximize the number of victims (Savage, Coogan & Lau, 2015). Some groups have begun to specifically target businesses seeking to cripple them, and in turn, seek a larger ransom (Arsene, 2016). Other groups, as appears to be the case with attacks against Linux, specialize in exploiting a specific technology. An example of this specialization would be the recent attacks targeting large data systems like CouchDB and Hadoop (Ragan, 2017). Because ransomware attacks are weakly targeted, all organizations are at risk. Specific software vulnerabilities, technologies in use, and defensive posture will elevate or lower the risk, but the odds of encountering a ransomware attack attempt is high with 47% of businesses reporting experiencing an attack between June of 2015 and June of 2016 (Osterman Research, Inc., 2016).

The most popular delivery mechanism for ransomware is phishing emails (Savage, Coogan & Lau, 2015). Attacks using this method usually target end-user systems and use social engineering to trick the system user into executing the malware. Alternate distribution methods have also been observed: Linux.encoder.1 was installed on systems by exploiting a weakness in the Magento e-commerce platform (Dr. Web, 2015). KeRanger was spread via modified copies of the Transmission BitTorrent client (Xiao & Chen, 2016). Samsam appears to have been installed manually or via a script from a

compromised server within the network (Beek & Furtak, 2016). This demonstrates that at least some ransomware gangs are flexible in which delivery mechanisms they are comfortable using. Organizations that take an e-mail centric approach to defense against ransomware may be caught off guard by these more innovative delivery approaches.

Ransomware authors have tried several different file encryption routines in their malcode. Many ransomware campaigns have been rendered ineffective by weak implementation of cryptography: Linux.encoder.1 is an example of this problem (Bitdefender Labs, 2015). More and more ransomware samples are using well-implemented strong cryptography, often using the operating system's built-in encryption libraries and commands. A key issue is whether the encryption is done before or after communicating with the command and control (C&C) infrastructure. Some ransomware families retrieve the encryption key used to encrypt the files from their C&C servers (Cyber Threat Alliance, 2015). Others encrypt the files and then transmit the encryption keys to their C&C servers when encryption is complete (Malwarebytes Labs, 2017). The issue of when the encryption is done is an important one as ransomware using the fetch-the-key-then-encrypt technique can be stopped by preventing communications with their C&C infrastructure..

It is common for ransomware to attempt to encrypt files on mapped drives and active cloud services, seeking to cause as much damage as possible (Krebs, 2016). Many ransomware samples will also go after mechanisms that allow for easy file recovery. Microsoft Windows offers the Volume Snapshot Service (VSS) which creates automatic backup copies of files. Windows ransomware samples will stop the VSS service and delete all current copies to prevent easy recovery of the system (Abrams, 2017).

Once the system is encrypted, the demand for ransom is issued. Bitcoin is by far the most popular payment option because it is anonymous by nature and criminals have developed a variety of techniques for laundering Bitcoins (Kotov & Rajpal, 2014). Other payment techniques have also been tried; the favorites tend to be online methods with weak or absent paper trails like Moneypak, Paysafecard and Ukash (Kharraz, Robertson, Balzarotti, Bilge & Kirda, 2015). Premium SMS numbers have also been used to some effect (Kharraz, Robertson, Balzarotti, Bilge & Kirda, 2015).

David Kennel, dakennel@gmail.com

After payment is rendered, victims may get their data back via a provided utility or set of keys and instructions. However, getting data back after payment is no guarantee, as there have been observed instances where issues with the malcode corrupted the data beyond recovery or where the attackers simply could not be bothered to provide the decryption keys. In a notable subset of the attacks against big data systems, the attackers delete the data and ask for ransom (Ragan, 2017). The demand indicates that they have backed up the data and will return it when the ransom is paid, while, in fact, the data has simply been deleted. The majority of victims do get their data back though because there is an economic incentive for the criminals to return it once paid (Krebs, 2016). Failure to return the data would result in victims assuming the data is lost and failing to pay the ransom, thus destroying the market for the attacks.

3.1 Notable Innovations in Ransomware Attack Patterns

Before diving into defensive strategies, a look at how ransomware attacks are likely to change in the future is valuable as this will also help tune defenses. Notable changes to the Ransomware attacks have been occurring in several areas. Ransom32 is the first sample of ransomware malcode to be implemented in a way that has an extreme potential for cross-platform use (Loeb, 2016). It is written in JavaScript using the NW.js[1] platform that allows for the development of platform native applications. Although presently targeted at the Windows operating system, this is a threat that could be trivially repackaged to attack Linux, Mac and Windows.

Attackers are also developing blended threats, either ransomware that steals data, or ransomware alongside a separate piece of malcode that steals data. The stolen data may be used in additional attacks, sold or used as part of the threat, e.g. “Pay us or not only will you lose your data, but we'll publish it.” CryPy is an interesting example of this type of malware; it uses a separate encryption key for each file, and sends the file names to the attacker. This creates the possibility of variable pricing for the return of the encrypted data (Ducklin, 2016).

1 See https://nwjs.io/ for more information.

There have also been some early indications of ransomware enhanced with worm-like propagation techniques. The Zcryptor malware characterized by Microsoft in May of 2016 drops autorun.inf files on removable storage devices which is a well-known technique for propagation of malware via removable drives (Kaspersky Lab, 2016). Since the returns to the criminals are largely driven by the number of infected hosts, defenders should anticipate that this type of propagation remains an active area of development by attackers.

The Linux/Unix threats seen to date are very different in terms of targeting and delivery from typical Windows ransomware samples. The overwhelming majority of ransomware targets Windows workstations and is delivered via phishing. There are four documented groups of attacks against Linux/Unix systems: Linux.encoder.1, KeyRanger, the big data attacks, and Samsam. None of them involve phishing attempts, a strong indicator that ransomware gangs recognize that, excepting MacOS, Unix/Linux systems are far more likely to be found in a server role. This also indicates that filtering for ransomware at the email gateway is a relatively weak control for protecting Unix/Linux systems as email is unlikely to be the attack vector.

The Linux.encoder.1 attacks leveraged a vulnerability in shopping cart and content management software from Magento (Dr. Web, 2015). Web merchants live and die by their store being available to customers and have been the targets of network DOS attacks in the past, making them logical targets for ransomware attacks. The Linux.encoder.1 attacks were ultimately unsuccessful due to a weakness in the implementation of the encryption routines (Bitdefender Labs, 2015). This attack indicates that attackers are able and willing to target systems for ransomware exploitation via server application weaknesses. This willingness to exploit application level weaknesses will be seen again in the discussion of the big data attacks.

The KeyRanger attack re-used the Linux.encoder.1 code on the MacOS platform, however, the packaging and path to exploitation were very different. In this case, the attack was packaged as a Trojan-infected version of the Transmission BitTorrent client, which is not particularly groundbreaking as tools and content commonly used for software and media piracy have been a hotbed for malware for years. What is interesting

is that the attackers subverted the legitimate Transmission download site to distribute their malware-infected version and that the malware had valid signatures that enabled it to bypass Apple's Gatekeeper defense (Xiao & Chen, 2016). These two techniques made the attack very difficult to defend against.

Over the past year there have been continuous attacks against a variety of big data processing systems (Cimpanu, 2017). These attacks have targeted a number of different big data applications including MongoDB, Redis, ElasticSearch, Hadoop, Cassandra and CouchDB (BinaryEdge, 2017). In these attacks, it appears that the attackers are simply exploiting poor configuration that has left these systems, or their API's, open to the Internet. These attacks are significant because of the targeting of big-data systems which typically contain a very large amount of critical enterprise data. These attacks also demonstrate the importance of developing and maintaining secure configuration baselines for production systems.

Samsam is an entirely different animal from the other Unix/Linux attacks. In this case the malware actually targeted Windows systems, but the attackers exploited a weakness in JBoss to gain access to the network, harvest information from Active Directory, and launch their attack against the accessible Windows systems (Hitchcock/Alert Logic Security Research, 2016). Samsam is notable in that the attackers exploited Linux/Unix systems but then chose to focus their attack against Windows hosts. The attack is worrying as it demonstrates that attackers are improving their cross-platform attack capabilities.

The Linux/Unix attacks seem to show that attackers recognize that they are more likely to encounter Linux and UNIX systems, with the exception of Mac, on the server side and are adjusting their tools and techniques accordingly. This server side targeting illuminates the major difference between Linux/Unix hosts and Windows when it comes to the ransomware threat.

David Kennel, dakennel@gmail.com

4. Ransomware Forecasting

It is difficult to assess how far an attack class and its related criminal ecosystem will grow. Over the past 10 years, malware of all types and the criminal ecosystems behind them have become more mature. Ransomware is a DOS attack against data; as such, those who are most likely to pay are those with weak or non-existent backups and those who are reliant on their data. Home users frequently fall into the first category and businesses fall into the second. Home users are unlikely to become religious about comprehensive data backups any time soon so ransomware authors are likely to continue to find success there. The fact that 47% of businesses reported ransomware attacks from June 2015 to June 2016 suggests that larger organizations too will continue to fall victim (Osterman Research, Inc., 2016).

Despite perennial statements that “this is the year of the Linux desktop”, Linux simply does not t have much of a presence on the desktop outside of a few small, mostly technical, user communities. For the foreseeable future, Apple's MacOS will remain the dominant desktop Unix flavor. Since the web browsing usage share of both Linux and MacOS is in the low single-digit percentage most of the end-user, focused ransomware development will continue to target the various flavors of Windows. The biggest danger for MacOS and Linux desktop users would be further extension of the Ransom32 malware into a true cross-platform threat, or other malware gangs picking up on the idea of using NW.js for cross-platform malcode development. It is reasonable to expect that ransomware attacks against systems being used as desktops will largely target MacOS as it represents the largest segment of desktop UNIX.

Netcraft operating system surveys suggest that Linux/Unix systems are the dominant operating system among web facing servers (Netcraft, 2017). It is reasonable to expect that this dominance on the web also means that there are a significant number of internally facing Linux/Unix servers in use by businesses. Servers typically contain vast troves of critical enterprise data, and high concentrations of data are prime targets for ransomware. The heavy presence of Linux/Unix systems in the server space indicates that there is a continued incentive for ransomware authors to pursue attacks against these systems.

David Kennel, dakennel@gmail.com

5. Defending Linux/Unix Systems

Kill chain analysis is a useful way to examine ransomware attack patterns and relevant Linux/Unix defenses. Kill chain analysis is a systematic approach to countering threats. Although it was originally developed for application to APT style attacks, the technique is broadly applicable (Hutchins, Cloppert & Amin, n.d.). Ransomware attacks can be characterized as follows:

1. Reconnaissance – Ransomware attacks usually do not feature extensive target reconnaissance. The normal model is to hit as many systems as possible to maximize financial returns. This approach may be changing as attacks against organizations show signs of increasing targeting.
2. Weaponization – Ransomware attacks are usually pre-weaponized in that the attackers have an exploit or social engineering angle and are using it against as many vulnerable systems as they can. As organizations improve their defenses. Attackers may respond in kind by tuning their attack to the targeted organization.
3. Delivery – Ransomware attacks are typically delivered via e-mail phishing but in the case of Linux/Unix systems, targeted attacks against server vulnerabilities are more often the rule.
4. Exploitation – In most cases, ransomware attacks exploit the human element via social engineering. Again, Linux/UNIX systems differ in that the attack pattern is typically server-focused with attacks exploiting application level vulnerabilities.
5. Installation – In a typical kill chain, this is the phase where an adversary would establish persistent access. Ransomware attacks are typically not interested in persistent access, although for blended threats there may be a component installed that permits on-going access.
6. Command and Control – Ransomware attacks are particularly reliant on the C&C phase for the exchange of keying material, although there have been innovations in this area that would permit the encryption to occur even if the C&C servers could not be reached.

7. Actions on Objectives – Ransomware attacks seek to deny access to data by encrypting it in a way that renders the data inaccessible to the owners and authorized users with the ultimate goal being to extort money from the victim. Blended ransomware threats may also exfiltrate data, either immediately or by dropping another Trojan. In the case of a two-stage attack like Samsam, the actions on objectives is the Active Directory data gathering and subsequent distribution of the second stage ransomware.

Ransomware attacks, because of the heavy automation and loose targeting, will typically pass through all seven stages of the kill chain in a very rapid fashion. Two-stage attacks such as Samsam are slower and require more interaction from the attackers.

Table 1 is a course of action matrix for ransomware style attacks that depicts enterprise level defenses and defenses specific to Linux/Unix platforms. This type of matrix customized to an organization is a useful way to put defenses in context and to ensure adequate defensive coverage.

Table 1: Ransomware action matrix

Phase	Detect	Deny	Disrupt	Degrade	Destroy
Reconnaissance Weaponization	Web activity analytics NIDS	Firewalls, router ACL NIPS			
Delivery Exploitation Installation	User detection User detection HIDS	Anti-virus at email gateway, Proxy malware filter Patch, user training, secure configuration Gatekeeper, Whitelist, SELinux	Host anti-virus Host anti-virus Web Application Firewalls Host anti-virus		
C&C	NIDS	Firewalls, router ACL	NIPS		

David Kennel, dakennel@gmail.com

Actions on objectives	User detection, Kernel audit logs	Gatekeeper, SELinux, Whitelist	Host anti-virus, Firewalls (outbound)	Backups	

Diving into the controls from Table 1, web activity analytics is a potentially useful technique to analyze activity on the corporate web presence. It can be a useful technique to identify a potential attacker who may be attempting to profile the company via open source information to discover prime phishing targets or publicly disclosed information about network architecture and defenses. This defense is of limited use against many ransomware strains as they are only loosely targeted, though it may reveal scans looking for specific weaknesses.

Network intrusion detection systems (NIDS) and network intrusion prevention systems (NIPS) are common in larger organizations and are within the reach of smaller organizations via Unified Threat Management (UTM) appliances that frequently bundle these capabilities. NIDS and NIPS systems have the ability to detect and/or block attacks based on network traffic. They can detect attacks via signature match, anomaly detection and protocol state analysis (Scarfone & Mell, 2007). Much like traditional anti-virus the weakness in these controls is that they are very reliant on signatures and attackers have developed techniques to evade them (Siddharth, 2010). Depending on its position in the network, NIDS/NIPS may get two opportunities to affect a ransomware attack; one, they may be able to identify or interdict the inbound exploit or second stage install as the exploitation of the host is occurring. They may also be able to see the C&C traffic as the malcode calls home to exchange keying material, though detection at this point means that a host is already executing the malware.

Secure configuration is the third of the CIS Top 20 Critical Security Controls, which notes in its discussion that, “the default configurations for operating systems and applications are normally geared to ease-of-deployment and ease-of-use – not security” (Center for Internet Security, 2015, p. 12). Secure configuration is a critical defense as demonstrated by the fact that the ransomware attacks against big data systems appear to be exclusively targeting insecurely configured applications. The availability of security

configuration guidance can vary widely within the Unix/Linux community. The US Defense Information Systems Agency (DISA) publishes secure configuration guidance for a variety of operating systems, as does the Center for Internet Security (CIS). A number of vendors have backed the Security Content Automation Protocol (SCAP) that is very useful for automated assessments and remediation of configuration settings. For Linux platforms, the OpenSCAP tool is available to perform scans using SCAP content. Unfortunately, production level SCAP content is generally only available for very popular distributions or ones where the vendor takes an active role. Typically, there is also a significant delay between a new version release and the release of the accompanying SCAP content. For specialized applications, like Hadoop, it is worth the time to carefully examine the application documentation to ensure that all access controls are correctly configured.

Vulnerability assessment and remediation, which is CIS Critical Security Control number four, is also critical. Prompt software patching has always been two (one?) of the best security defenses. Linux.encoder.1 targeted a software weakness in shopping cart software for which patches had been available for 8 months (Dr. Web, 2015). The Samsam attacks targeted known vulnerabilities in JBoss to gain a foothold in the network before exploiting the hosts inside the network (Hitchcock/Alert Logic Security Research, 2016).

There are a few tools that can help with both secure configuration assessment and vulnerability assessment. SCAP, if used in concert with its Open Vulnerability and Assessment Language (OVAL), is capable of performing vulnerability assessments of hosts. Another approach is network-based vulnerability scans like those done by OpenVAS or the Nessus vulnerability scanner. With either technique, the best practice is to routinely scan systems for issues and to deploy automated tools for patching and configuration management (Center for Internet Security, 2015).

Training to help users recognize and respond appropriately to phishing attempts is as valuable of a defense against ransomware as exploiting the human element is a common vector. Users also need to be trained to recognize the ransomware attack pattern, since, due to the speed of a typical ransomware attack, they may be able to report the

incident to the response team before the team can pick it out of the alerts from other sensors.

Many Unix/Linux system administrators have questioned the need for anti-virus or anti-malware software on Linux/Unix systems due to the very low number of Linux/Unix specific malcode threats (Goretsky, 2015). Attackers are aggressively repacking and obfuscating malware to avoid detection by anti-virus software (Krebs, 2014). Despite these issues, anti-virus/anti-malware still provides some value as part of a layered defense. Firstly, it provides some detection and prevention at the signature and heuristic level (depending on the specific anti-virus software used) to protect the Linux/Unix host itself. It also helps to interfere with attackers looking to use a Linux/Unix host to pivot into infecting the Windows systems on the network as was seen in the Samsam attack campaign (Hitchcock/Alert Logic Security Research, 2016).

There are two key choke points for most organizations where the dominant ransomware attacks can be stopped: e-mail servers and proxy servers. Anti-malware software, reputation filters and content filters deployed at these points can provide vital protection to hosts on the network. In the case of Unix/Linux specific ransomware attacks this protects primarily MacOS, as it is the most likely to be used in a client role, and the most likely to be targeted as a client by attackers. Since other UNIX/Linux systems are likely to be targeted in their role as servers, not as web or e-mail clients, these controls may be ineffective for defending those systems depending on their location within the network.

There are a number of security tools available to Linux/Unix web, email and proxy servers. The Squid proxy server is supported by an array of content filters including DansGuardian, SquidGuard and ufdbGuard. Squid also supports the Internet Content Adaptation Protocol (ICAP) which allows traffic to be routed through an anti-malware scanner or a content filtering appliance. Linux/Unix email servers are usually based on Sendmail or Postfix. Both of these mail servers can be extended to support sender policy framework (SPF)[2], domainkeys identified mail (DKIM)[3] and domain-based

2 http://www.openspf.org/

message authentication, reporting and conformance (DMARC)[4] which can filter out email with forged sender addresses. A variety of spam filters and anti-virus options may also be integrated into the mail processing chain. These proxy and mail security tools help protect the clients using the services provided by these systems. To protect the servers, defenders should look again to NIDS/NIPS solutions.

Web servers can be tricky to defend because the applications deployed on them create additional attack surface. One way to help control this is by deploying a web application firewall (WAF) which can be used to monitor and limit the types of traffic allowed to pass through to the application. Dedicated WAF appliances are available from a number of vendors and an open source tool called ModSecurity[5] is available for the Apache and Nginx web servers.

Whitelisting, also known as Application Control, software is often promoted as an alternative for malware control and prevention. The concept is simple: instead of enumerating the potentially infinite realm of malicious software, enumerate the much smaller set of known good software. A few vendors provide Whitelist software solutions that support some Linux distributions. Whitelisting is a powerful tool that does have some limitations. Depending on the tool and configuration, whitelisting software may not control all of the software on the system. Commonly omitted code includes Java and kernel modules that have been leveraged in attacks in the wild (Shackleford, 2009). Whitelisting can also create challenges for the change management process, as updates to software may need to be pre-approved in the whitelist solution before deployment to production. Even with these limitations, whitelisting creates a significant barrier for attackers.

On MacOS, Apple has implemented a security feature known as Gatekeeper that requires that all programs be signed or authorized by the system administrator before

---

3 http://www.dkim.org/

4 See https://dmarc.org/ for more information.

5 See https://modsecurity.org/ for more information.

being allowed to execute (Apple, Inc., 2016). This is essentially a whitelist defense as described above. Unfortunately, attackers have already demonstrated the means to bypass this defense by acquiring valid signing keys as was observed in the KeyRanger attack (Xiao & Chen, 2016).

An important defensive tool available to Linux systems is SELinux. SELinux is a kernel-level security feature that provides a Mandatory Access Control (MAC) mechanism. Policies can be defined that limit the capabilities and data access rights of users and processes. As a simple example of how SELinux works, suppose that an administrator accidentally changed the mode of the /etc/shadow file on a Red Hat Enterprise Linux 7 system to 644, rendering it world readable. If the SELinux system is in enforcing mode then the Apache web server process & user would not be able to read the /etc/shadow file because the SELinux Targeted policy shipped with Red Hat Enterprise Linux prevents that access, even though the file permissions would seem to allow it. A well-tuned SELinux policy can help prevent exploitation and damage to data by controlling service permissions.

There are a few different approaches to host intrusion detection systems (HIDS) on Linux/Unix systems. Tripwire and the Advanced Intrusion Detection Engine (AIDE) are file and directory monitoring tools that use file metadata and checksums to monitor for changes. These tools typically do not alert in real time and are therefore of limited value in detecting a ransomware attack which is likely to have encrypted the data before the reports are read. They do have value in detecting blended threats that are dropping a secondary component in addition to the ransomware. A second, more responsive approach can be seen in tools like OSSSEC that monitor multiple aspects of the system and are capable of raising alerts in or near real time. Due to the speed of a ransomware attack, this may or may not prevent damage to the infected system. In the case of ransomware exhibiting worm characteristics, it may give responders an opportunity to prevent spread to other hosts by isolating the victim system.

Some Unix/Linux systems have a kernel auditing mechanism. Auditd is the Linux version of this capability. Kernel audit logs can be very detailed and can log activity that does not show up in the normal UNIX system logs. Auditd even has the ability to send

the kernel audit logs to another machine for backup, correlation and reporting. These audit records can provide another means to detect a ransomware in progress. Unless the logs are being centrally collected and analyzed in real time, they will only be of use during the forensic analysis of the attack after the fact.

While firewalls are easily bypassed via phishing attacks by most ransomware samples, for Unix/Linux systems, which are likely to be targeted as servers, they hold much more value by limiting the avenues available for attack. There are a number of different lists of IP addresses of known or suspected bad hosts on the Internet. These blacklists can be used at the host- and/or network- level firewalls to prevent traffic from reaching these suspect hosts. Attention should be paid to outbound filtering via firewalls, as this is a valuable control in the case of ransomware because many samples must fetch a key from their command and control (C&C) servers before they will encrypt data. A firewall that prevents the outbound communication with the C&C servers may stop a ransomware attack before it has an opportunity to do damage to the system.

5.1 Smart Backup & Restore

Data backups are the best and ultimate line of defense against ransomware attacks. Perhaps the biggest issue that the growth of ransomware creates is the need to re-evaluate backup policies and procedures. For many organizations, the backup and restore processes are designed around the assumption that restoration of a full system is a rare event. Ransomware greatly increases the potential for a full system, or multi-system restoration that means that organizations must re-evaluate whether their backup intervals, processes and mean time to restoration (MTTR) are appropriate given the increased likelihood that emergency procedures will be called upon. The MTTR consideration is especially important, as one of the major impacts of a ransomware attack is the downtime that it causes. Hollywood Presbyterian Medical Center paid the ransom and still suffered ten days of impaired operations (Wagner, 2016).

To be an effective defense against ransomware, backups must be inaccessible to the malware which is something that tape-based backup excels at. Thanks to the falling prices of disk and the speed of data access for single file or single folder recovery, many

organizations have switched to disk-based backup solutions. Backing data up to a mounted NFS export from another host would normally be a valid form of backup. However, many of the ransomware families for Windows make a point of encrypting data available on all mapped drives, including network shares. This problem is actually worse on most Linux/Unix systems as mounted file systems are transparently integrated into the file system hierarchy, meaning that ransomware does not have to actively look for mounted external drives or NFS exports.

A potentially mitigating factor, but one that should not be relied upon to any great degree, is that on recent versions of NFS the root user has sharply limited permissions on mounted NFS file systems. This means that the malcode actually would do less damage to a mounted NFS file system running as root than it would running as an unprivileged user. This is a good reason to avoid the "no_root_squash" option when configuring NFS as that option gives root processes full control over files in the NFS exported space.

The problem with backups as a defense against ransomware is the downtime required to restore the affected systems from backup. For systems that have very large stores of data, the restore time can be quite long, resulting in significant downtime and business impact. Regardless of the backup technology used, tape, disk or cloud, organizations should test their recovery processes and ensure that they can restore their system quickly and correctly. For organizations with high up-time requirements, a potentially useful enhancement to backups is to use rigorous automated change management commonly used in DevOps environments (e.g. Ansible, Chef, Puppet, FAI, etc.) combined with PXE boot and the ability to re-image systems via automation. A well-designed solution would allow a server farm to be re-imaged and restored to operational status automatically in a short period.

Database servers frequently contain the crown jewels of corporate data and require special consideration when designing a backup and restore procedure. On-line transaction processing systems (OLTP) like web stores or customer service portals are very likely to lose transactions in a ransomware type attack unless care has been taken to continuously backup the transaction logs elsewhere.

David Kennel, dakennel@gmail.com

Backup and restore procedures should be regularly tested for reliability to ensure that the process still satisfies the MTTR targets. System inventories should also be compared to the list of systems being backed up on a regular basis to ensure that new systems are adequately backed up. Backup is the best defense against ransomware, but it is also the last line of defense. If it fails, organizations may end up having to pay the ransom or suffer data loss.

6. Conclusion

This paper bases its claims on data from industry and scholarly research into crypto-ransomware attacks. Source information includes summary papers, multi-family ransomware analysis papers, deep dive analysis of specific samples and news reporting regarding ransomware campaigns. While ransomware attacks against Windows targets generally rely on phishing and social engineering, attacks against Unix/Linux systems are more likely to exploit poor configuration or remotely exploitable vulnerabilities. Ransomware authors have been steadily improving their use of cryptography and increasing their targeting of businesses. Unix/Linux defenders do have a variety of different defenses available to them including; NIDS, SELinux, secure configuration, firewalls, automated configuration management and backups which if well designed can help prevent ransomware attacks from being successful or can minimize the attack's business impact.

Clearly ransomware attacks against, or leveraging, Unix/Linux systems are more than just theoretical. The concentration of data on Unix/Linux servers will likely serve as sufficient inducement for attackers to continue targeting these systems for these types of attacks. Fortunately, there are viable defenses against ransomware attacks. While many organizations may already have many of these defenses in place, it is important that they are checked and re-tuned to ensure proper response against the increasing ransomware threat.

David Kennel, dakennel@gmail.com

References

Abrams, L. (2017, January 31). CryptoMix variant named CryptoShield 1.0 ransomware distributed by exploit kits. Retrieved from https://www.bleepingcomputer.com/news/security/cryptomix-variant-named-cryptoshield-1-0-ransomware-distributed-by-exploit-kits/

Apple, Inc. (2016, March 23). OS X: About Gatekeeper. Retrieved from https://support.apple.com/en-us/HT202491

Arsene, L. (2016). Corporate ransomware attacks and what to expect next. Retrieved from https://www.rsaconference.com/blogs/corporate-ransomware-attacks-and-what-to-expect-next

Beek, C., & Furtak, A. (2016). Targeted ransomware no longer a future threat. Retrieved from Intel Security website: http://www.intelsecurity.com/advanced-threat-research/content/Analysis_SamSam_Ransomware.pdf

BinaryEdge. (2017, January 18). The compendium of database ransomware. Retrieved from http://blog.binaryedge.io/2017/01/18/the-compendium-of-database-ransomware/

Bitdefender Labs. (2015, November). Linux ransomware debut fails on predictable encryption key. Retrieved from https://labs.bitdefender.com/2015/11/linux-ransomware-debut-fails-on-predictable-encryption-key/

Bitdefender Labs. (2016, March). KeRanger is actually a rewrite of Linux.encoder. Retrieved from https://labs.bitdefender.com/2016/03/keranger-is-actually-a-rewrite-of-linux-encoder/

Center for Internet Security. (2015). Critical security controls for effective cyber defense (Version 6.0). Retrieved from https://www.cisecurity.org/critical-controls.cfm

Cimpanu, C. (2017, January 18). Database Ransom Attacks Hit CouchDB and Hadoop Servers. Retrieved from

https://www.bleepingcomputer.com/news/security/database-ransom-attacks-hit-couchdb-and-hadoop-servers/

Cyber Threat Alliance. (2015). Analysis of the cryptowall version 3 threat. Retrieved from http://cyberthreatalliance.org/cryptowall-report-v3.pdf

Dr. Web. (2015, November 6). Encryption ransomware threatens Linux users — Dr.Web - innovative anti-virus technologies. Comprehensive protection from Internet threats. Retrieved from http://news.drweb.com/show/?i=9686&c=5&lng=en&p=0

Ducklin, P. (2016, October 18). Data-stealing CryPy ransomware raises the specter of variable pricing for files – Naked Security. Retrieved from https://nakedsecurity.sophos.com/2016/10/18/data-stealing-crpy-ransomware/

Goretsky, A. (2015, January 13). Do you really need antivirus software for Linux desktops? Retrieved from http://www.welivesecurity.com/2015/01/13/really-need-antivirus-software-linux-desktops/

Hitchcock/Alert Logic Security Research, J. (2016, September 23). SamSam Ransomware. Retrieved from https://www.alertlogic.com/blog/samsam-ransomware/

Hutchins, E., Cloppert, M., & Amin, R. (n.d.). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Retrieved from Lockheed Martin Corporation website: http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf

Kaspersky Lab. (2016, July 14). Crypto-Ransomware Attacks are Now Targeting Corporate Users. Retrieved from http://usa.kaspersky.com/about-us/press-center/press-releases/2016/Crypto-Ransomware-Attacks-are-Now-Targeting-Corporate-Users

Kaspersky Lab. (2016, June 2). ZCryptor: The conqueror worm – Kaspersky Lab official blog. Retrieved from https://blog.kaspersky.com/zcryptor-ransomware/12268/?slow=1

David Kennel, dakennel@gmail.com

Kelion, L. (2013, December 24). Cryptolocker ransomware has 'infected about 250,000 PCs' - BBC News. Retrieved from http://www.bbc.com/news/technology-25506020

Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015). Cutting the gordian knot: A look under the hood of ransomware attacks. Retrieved from https://wkr.io/publications/dimva2015ransomware.pdf

Kotov, V., & Rajpal, M. (2014). Understanding crypto-ransomware: In-depth analysis of the most popular malware families. Retrieved from Bromium, Inc. website: https://www.bromium.com/sites/default/files/bromium-report-ransomware.pdf

Krebs, B. (2014, May 7). Antivirus is dead: Long live antivirus! Retrieved from https://krebsonsecurity.com/2014/05/antivirus-is-dead-long-live-antivirus/

Krebs, B. (2016, March 22). Hospital Declares 'Internal State of Emergency' After Ransomware Infection — Krebs on Security. Retrieved from https://krebsonsecurity.com/2016/03/hospital-declares-internet-state-of-emergency-after-ransomware-infection/

Krebs, B. (2016, January 14). Ransomware a Threat to Cloud Services, Too — Krebs on Security. Retrieved from https://krebsonsecurity.com/2016/01/ransomware-a-threat-to-cloud-services-too/

Loeb, L. (2016, January 4). Cross-Platform Cryptoware Is Here. Retrieved from https://securityintelligence.com/news/cross-platform-cryptoware-is-here/

Malwarebytes Labs. (2017, January 31). Locky Bart ransomware and backend server analysis. Retrieved from https://blog.malwarebytes.com/threat-analysis/2017/01/locky-bart-ransomware-and-backend-server-analysis/

Netcraft. (2017, February). January 2017 web server survey. Retrieved from https://news.netcraft.com/archives/2017/01/12/january-2017-web-server-survey.html

David Kennel, dakennel@gmail.com

Osterman Research, Inc. (2016). Malwarebytes | Osterman Survey: Understanding the Depth of the Ransomware Problem in the United States. Retrieved from https://www.malwarebytes.com/surveys/ransomware/

Ragan, S. (2017, January 3). Exposed MongoDB installs being erased, held for ransom | CSO Online. Retrieved from http://www.csoonline.com/article/3154190/security/exposed-mongodb-installs-being-erased-held-for-ransom.html

Ragan, S. (2017, January 30). MongoDB ransom attacks continue to plague administrators | CSO Online. Retrieved from http://www.csoonline.com/article/3162711/security/mongodb-ransom-attacks-continue-to-plague-administrators.html

Savage, K., Coogan, P., & Lau, H. (2015). Security response: The evolution of ransomware. Retrieved from Symantec website: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-evolution-of-ransomware.pdf

Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS) (800-94). Retrieved from National Institute of Standards and Technology website: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf

Shackleford, D. (2009). Application whitelisting:Enhancing host security. Retrieved from SANS website: https://www.sans.org/reading-room/whitepapers/analyst/application-whitelisting-enhancing-host-security-34820

Siddharth, S. (2010, November 2). Evading NIDS, revisited | Symantec Connect. Retrieved from http://www.symantec.com/connect/articles/evading-nids-revisited

Solomon, A., Nielson, B., & Meldrum, S. (n.d.). aids.tech.info. Retrieved from http://ftp.cerias.purdue.edu/pub/doc/general/aids.tech.info

Wagner, L. (2016, February 17). LA hospital pays hackers nearly $17,000 to restore computer network. Retrieved from http://www.npr.org/sections/thetwo-

way/2016/02/17/467149625/la-hospital-pays-hackers-nearly-17-000-to-restore-computer-network

Xiao, C., & Chen, J. (2016, March 6). New OS X Ransomware KeRanger Infected Transmission BitTorrent Client Installer - Palo Alto Networks Blog. Retrieved from http://researchcenter.paloaltonetworks.com/2016/03/new-os-x-ransomware-keranger-infected-transmission-bittorrent-client-installer/