Detecting Ransomware with Honeypot techniques

Chris Moore
Computing and Media Services
University of St Mark & St John
Plymouth, England
e-mail: cmoore@marjon.ac.uk

***Abstract*— Attacks of Ransomware are increasing; this form of malware bypasses many technical solutions by leveraging social engineering methods. This means established methods of perimeter defence need to be supplemented with additional systems. Honeypots are bogus computer resources deployed by network administrators to act as decoy computers and detect any illicit access. This study investigated whether a honeypot folder could be created and monitored for changes. The investigations determined a suitable method to detect changes to this area.**

This research investigated methods to implement a honeypot to detect ransomware activity, and selected two options, the File Screening service of the *Microsoft File Server Resource Manager* feature and *EventSentry* to manipulate the Windows Security logs. The research developed a staged response to attacks to the system along with thresholds when there were triggered. The research ascertained that witness tripwire files offer limited value as there is no way to influence the malware to access the area containing the monitored files.

Keywords-honeypot; ransomware; malware; computer security; cyber security, network, detect, activity

OVERVIEW AND PROBLEM STATEMENT

Cyber-extortion [1] methods can be traced back to the late 1980s. However the reports of the modern wave of ransomware begin in 2005 [2], this is a form of malicious software (malware) that hackers are spreading with the intent of not just destroying data as traditional attacks; but encrypting and charging for the service to recover the data. Ransomware is variety of scareware, this is when the user is prompted to pay the ransom in reaction in fear of losing their data. This occurrence of this variation of malware is on the increase [3]. This is a profitable business model for the criminal organisations that orchestrate the attacks. Payment is often via bitcoin, with requests in the region of 500 – 1000 USD. This 'affordable' price is set to increase as time goes on, making it look attractive to pay early. This is a lucrative business for the cybercriminals, one calculation [4] suggests the figure of 200 million USD per year is extorted by the criminal gangs. Advice is often given not to pay the ransom, as this perpetuates the criminal business model, however it may be the only way to recover lost data.

As the ransomware threat evolves, different varieties of the malware progress, some names are well known, such as CryptoLocker and CryptoWall, in recent times TeslaCrypt and Locky have appeared.

Detection of malware before it begins its payload of encrypting files is difficult; traditional antivirus products need to collect a sample of malware, analyse it, and deploy the updates to the virus signature files to the protected machines, Figure 1. illustrates how after several days of a new attack being circulated, only half of anti-virus vendors provide protection for this attack.

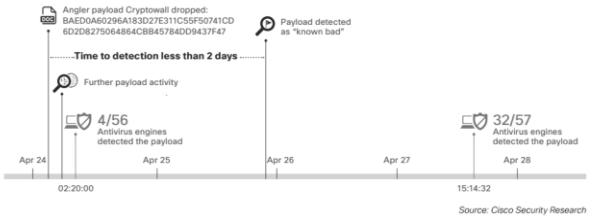

Figure 1. Time to detect malware [5]

As traditional antivirus software cannot detect new forms of malware quickly enough to protect systems, there is a need to detect when ransomware begins to activate and block further damage.

A possible solution could be a honeypot [6], that is a computer system deployed specifically to detect unauthorised use of a resource. As the honeypot system does not expect any legitimate connections, any interaction would be considered as an attack on the system. This information would be used to raise an alert of an attack

This research looks into how honeypot principles could be utilised to detect and possible mitigate a ransomware attack on a *Microsoft Windows* network.

LITERATURE SURVEY

Microsoft [7] advice on tackling ransomware is that a tested reliable backup regime is the best way to mitigate the damage from a ransomware attack. While antivirus is still advocated, as we have seen, this may not be updated soon enough to block an attack. Microsoft also suggest AppLocker to block programs from running in common places, this is good advice for computers on a managed domain. However, there is still a possibility of new variants

of malware writing to uncontrolled areas. This is why this investigation has evolved.

Detecting ransomware is difficult due to its morphing nature; it has already evaded perimeter defence on the firewall or spam filter. There is no simple signature to look for to indicate the presence of ransomware. Many use an extension *.locky*, but the malware evolves and could easily be *.encrypted* or *.nochance* depending on the variant. Detection would depend on an updated list of filename patterns [8] which would be onerous for the network administrator to keep up to date. Therefore, looking for specific filenames or extensions as evidence of an attack was rejected as a suitable method. An alternative method [9] proposed a machine learning based system. This approach looked for threatening text associated with a ransom note, along with analysing data flows to determine if encryption is taking place. Unfortunately, this solution was for the *Android* platform, and would not be transferable to the required *Windows* platform.

Knowledge of how to detect activity in a ransomware would be a central need to this research. Kharraz, Amin, et al. [10] advise monitoring the Master File Table (MFT) for activity, and also suggest decoy resources a method of detecting activity.

First, looking for a commercial solution to the problem, *Varonis* in their *DatAdvantage* product use User Behaviour Analytics (UBA) [11] to determine baseline normal activity. Later, when abnormal activity occurs, such as thousands of file modifies in a short time, this can trigger an email the alerting the administrator and user that unusual access has occurred. Another commercial product is HitmanPro [12] which detects unusual system behaviour, rather than typical static anti-virus signatures. A HitmanPro feature to share detected activity with VirusTotal gives the opportunity to learn more about the attacks.

A second approach looked at placing some key files across the network and monitoring for changes. This tripwire idea [13] & [14] utilise witness files that were monitored for modification or deletion. If a witness file is tampered with, the Lanman server service is stopped. Figure 2. illustrates a rudimentary script to perform a similar task, that if a difference is detected in a copy of file to the original, network services will be stopped.

```
if ((Get-FileHash .\ReadMe.txt).hash -ne (Get-FileHash .\original\ReadMe.txt).hash )
        {Stop-Service "LanmanServer" -force}
```

Figure 2. Minimal Powershell script detect and react to a change

A third method of detecting changes was to utilise a function built into *Windows Server 2012*, File Server Resource Manager (FSRM). The suggestion of a canary resource [15] takes its name from the practice of coal miners taking canaries down mines as an early warning against toxic gases. A function to control access is called File Screening, and can be used to block the writing of unauthorised files. This idea was expanded [16] to utilise PowerShell to block an offending user's access.

The fourth and final solution follows the blogs from EventSentry [17] over the past 3 years, with the EventSentry product being able to monitor *Windows* Security logs, and trigger actions when user activity passes a threshold. This is a Security information management (SIM) product to aggregate log files. Actions can be to send an email or invoke a server shutdown. This product is available as full featured commercial product, however the free *EventSentry Light* provides the functionality to undertake necessary monitoring and action

Having determined there were multiple approaches to detect ransomware, further investigation was required to establish which solution would best meet our requirements to invoke an action on the discovery of an intrusion.

Experiment Requirements

At some point, prevention methods will not be able to defend against new and unknown attack techniques, therefore the next line of defence arises from intrusion detection systems. Looking to use a honeypot as an intrusion detection system, honeypots do not prevent intrusions [18], but comparable to a burglar alarm where an indicator of an intrusion gives the system administrator an opportunity to prevent any further spread of damage to the system.

The intention of the research was to determine a suitable ransomware detection method and deploy this to add an additional layer of security to the network; to protect the network actions must be taken on the knowledge of an attack, nevertheless shutting down a server when a user legitimately updates a collection of files would be a severe response. Conversely, not reacting quickly to a ransomware attack would result in more files becoming encrypted.

To moderate against non-malware usage triggering overly harsh actions, a hierarchy of responses was identified. The model for responding to alerts is shown in Figure 3.

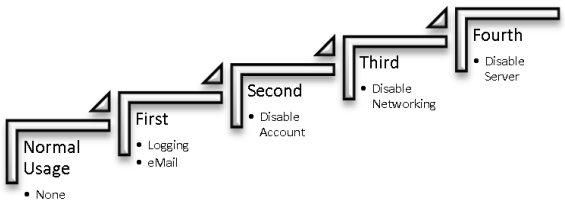

Figure 3. Tiered response to detection

A. First

Triggered when the first detection of a change has been encountered. This was to look to email the system administrator that changes had been made to the monitored folder.

B. Second

Generated when more activity of has been encountered. This level looked to determine the username or station name of the attacking malware. With this information the user could have their network account disabled or the station sessions disconnected. This would only impact the necessary user, unfortunately once the network resource has been connected the status of the account

C. Third

The next intensity activity would propose to stop the network services.

D. Fourth

Produced when a fourth threshold of changes has been encountered. At this point it would be determined that alerting the administrator and blocking the users access was not enough to stop the spread of damage, therefore the ultimate protection would be to shutdown the server.

SELECTION OF METHOD

Trails placing witness files scattered across the network did not attract any ransomware activity in the period of the study. Analysis of ransomware actions indicated that the attack often would progress alphabetically through mapped drives, therefore a development to the trail was to map an early letter of the alphabet to the honeypot area. While a filename such as #######Tripwire.txt would be alphabetically one of the first encountered, this would easily be thwarted by reversing the order of files attacked, meaning any early detection files would be attacked last. Consequently the use of a witness file as a tripwire to detect activity was eliminated.

In a similar way user behaviour analytics were also rejected, once UBA has learnt normal activities, would detect extraordinary access to a storage area such as rogue users or a compromised account. Perhaps a short trail of only a month with UBA to detect ransomware was not long enough to learn behaviour correctly, nevertheless a protection system that was unable to provide security while awaiting the baseline collection did not provide confidence that forecasted attacks would be intercepted sometime in the future.

This leaves our investigation with two approaches to detecting ransomware, initially, a honeypot folder monitored with a FSRM File Screen, followed by observing changes to the Windows Event Logs.

The FSRM followed the guidance in [16] and can be updated with known filename and extensions of latest attacks hosted on GitHub [19]. This is an effective method to block ransomware being written to a specific honeypot folder.

Next, EventSentry was configured following the instructions [17] to set up file auditing to event *4663: An attempt was made to access an object*. Actions were setup to follow the three tiers, email, Stop Server service and finally shutdown the service. These would be linked to filters, with the required thresholds to trigger the action. Determining this threshold needs some consideration, to low, and many false alerts would be generated, conversely, too high, would result in never triggering. Each network will exhibit different usage characteristics, but for the experiment, a ten second period was considered. In the experimental set up normal activity was monitored and averaged over a day. Double the normal activity was the baseline for initiating any action, therefore over 50 file changes elevate to first tier, three times the baseline, that is 150 would elevate to the second tier, and ten times the baseline, 500, would trigger the third tier, finally 1000 would activate tier four. The ability to copy and paste the filters and amend the thresholds allowed this process to be completed efficiently. Figure 4. shows a threshold configuration screen for EventSentry, the GUI allowing straightforward edits of the experimental systems to be made without needing to enter complex command line statements.

With the honeypot experiment setup, it was ready for attack, while a live attack would test our defences, it would be reckless to invite this, therefore a simulated ransomware script [20] was employed.

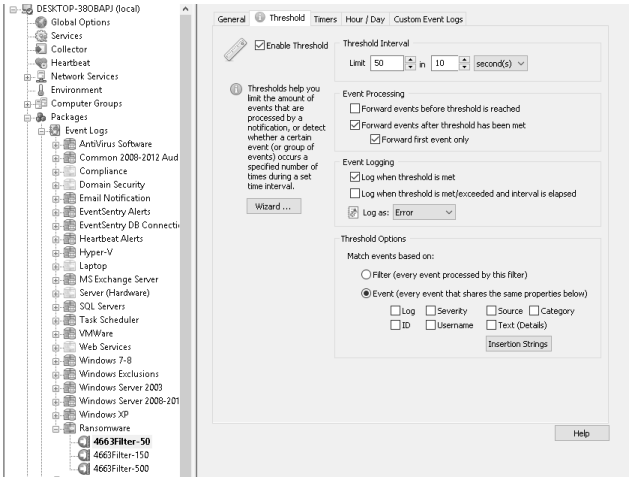

Figure 4. EventSentry Filter Threshold Configuration

EXPERIMENTAL DATA AND RESULTS

During the experimental period, no live ransomware was encountered by the honeypot, this can be attributed to domain wide Applocker control and current user awareness ransomware. However the simulated activates were able to assess the performance of the experiment, first from FSRM, Figure 5. shows an early warning the email sent to the

administrator and user that unauthorised access has taken place.

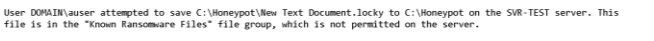

Figure 5. Email alert from FSRM

Increasing the number of files in the simulated ransomware activities also allows checking when the network server service was disabled. Figure 6. shows a screen shot of a how the network activity is interrupted when the activity level is detected.

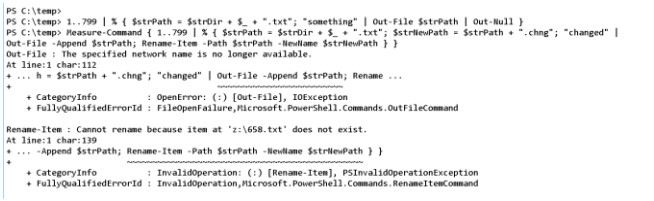

Figure 6. Error messages show contact with server has been stopped

The simulations indicate how many files can be attacked in a short period, but also that in the case of stopping the Lanman Service, this was found to have stopped access in six seconds after the start of the simulation.

CONCLUSION AND FURTHER WORK

Further simulated testing can be provided with a controlled deployment of the Hidden Tear [21] project.

While it is possible to deploy honeypot type fake folders with tripwire files for ransomware to interact with, the nature of the decoy folders is that there is no guarantee the malware would attempt to invade these areas, and therefore bypassing this defence. This limited view of a system is a disadvantage of honeypots, as a honeypot free from attack alerts is not an indicator that other areas are not being targeted. As malware is automated and will target any location arbitrarily, placement of a honeypot anywhere to detect activity is an improvement on no monitoring at all. Honeypot principles around collecting information about an attack and using it for defence are still valuable, the study has shown the honeypot can identify the user along with the volume of files being modified and this can inform actions. Email alerts to users also need to be accompanied with the user awareness training, possibly the message needs to request the network cable is unplugged.

Most alarming at the point in time when this paper was written, Ransomware moved up a gear, as to begin with Ransomware was delivered as a Trojan, but as of late May 2016, reports of ZCryptor [22] emerge that state this variant also replicates it code onto removable and network drives.

ACKNOWLEDGMENT

Thanks to Ameer Al-Nemrat, University of East London for encouragement on developing this paper and support from the Computing and Media Services team at the University of St Mark & St John. Finally, thank you to the editors and peer reviewers for their time, expertise and guidance on this paper

REFERENCES

[1] N. Hampton and Z. A. Baig, “Ransomware: Emergence of the cyber-extortion menace,” in *Australian Information Security Management*, Perth, 2015.

[2] A. Gazet, “Comparative analysis of various ransomware virii,” in *Journal in Computer Virology*, 2008, pp. 77-90.

[3] M. Garnaeva , J. van der Wiel , D. Makrushin , A. Ivanov and Y. Namestnikov , “Kaspersky Security Bulletin 2015. Overall statistics for 2015,” Kaspersky Labs, 15 December 2015. [Online]. Available: https://securelist.com/analysis/kaspersky-security-bulletin/73038/kaspersky-security-bulletin-2015-overall-statistics-for-2015/. [Accessed 21 May 2016].

[4] C. Everett, “Ransomware: to pay or not to pay?,” *Computer Fraud & Security,* vol. 4, pp. 8-12, 2016.

[5] Cisco, “Cisco 2015 Midyear Security Report,” Cisco, San Jose, 2015.

[6] L. Spitzner, Honeypots: tracking hackers, Boston: Addison Wesley, 2002.

[7] D. Mauser and K. Cenerelli, “Microsoft Protection Center: Security Tips to Protect Against Ransomware,” 6 April 2016. [Online]. Available: http://social.technet.microsoft.com/wiki/contents/articles/29787.microsoft-protection-center-security-tips-to-protect-against-ransomware.aspx. [Accessed 2 June 2016].

[8] J. Ned, “List of ransomware extensions and known ransom files created by Crypto malware,” 16 February 2016. [Online]. Available: https://www.reddit.com/r/sysadmin/comments/46361k/list_of_ransomware_extensions_and_known_ransom/. [Accessed 2 June 2016].

[9] N. Andronio, S. Z. Zanero and F. Maggi, “Heldroid: Fast and Efficient Linguistic-Based Ransomware Detection,” *Research in Attacks, Intrusions, and Defenses,* vol. 9404, pp. 382-404, 21 October 2015.

[10] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge and E. Kirda, “Cutting the gordian knot: a look under the hood of ransomware attacks,” in *Detection of Intrusions and Malware, and Vulnerability Assessment* , Springer International Publishing, 2015, pp. 3-24.

[11] C. Ng, “The Complete Ransomware Guide,” Varonis, 17 December 2015. [Online]. Available: https://blog.varonis.com/the-complete-ransomware-guide/. [Accessed 2 June 2016].

[12] SurfRight, “HitmanPro,” 2015. [Online]. Available: http://www.surfright.nl/en/home/. [Accessed 8 July 2016].

[13] Ben22, “Cryptolocker - using Powershell as a tripwire,” Reddit, 12 November 2013. [Online]. Available: https://www.reddit.com/r/sysadmin/comments/1qf7yi/cryptolocker_using_powershell_as_a_tripwire/. [Accessed 3 June 2016].

[14] Appleton, Alex, “Download CryptoLocker Tripwire 1.0,” 24 April 2015. [Online]. Available: http://alexappleton.net/post/83785313416/download-cryptolocker-tripwire-10. [Accessed 14 May 2016].

[15] J. Dale, “Cryptolocker Canary - detect it early!,” 21 November 2014. [Online]. Available: https://community.spiceworks.com/how_to/100368-cryptolocker-canary-detect-it-early. [Accessed 13 May 2016].

[16] Netwrix, “Ransomware Protection Using FSRM and PowerShell,” 11 April 2016. [Online]. Available: http://blog.netwrix.com/2016/04/11/ransomware-protection-using-fsrm-and-powershell/. [Accessed 16 May 2016].

[17] I. Koecher, “Defeating Ransomware with EventSentry & Auditing,” 2 March 2016. [Online]. Available: http://www.eventsentry.com/blog/2016/03/defeating-ransomware-with-eventsentry-auditing.html. [Accessed 16 May 2016].

[18] W. Stallings, Network Security Essentials: Applications and Standards, Harlow: Pearson, 2013.

[19] thephoton, “ransomware,” 10 April 2014. [Online]. Available: https://github.com/thephoton/ransomware. [Accessed 17 June 2016].

[20] T. Rayner, “Simulating A Ransomware Attack With PowerShell,” 27 January 2016. [Online]. Available: https://blogs.technet.microsoft.com/canitpro/2016/01/27/simulating-a-ransomware-attack-with-powershell/. [Accessed 13 May 2016].

[21] U. Sen, “An open source ransomware honeypot,” 2016. [Online]. Available: https://github.com/utkusen/hidden-tear. [Accessed 8 July 2016].

[22] Microsoft, “Link (.lnk) to Ransom,” 26 May 2016. [Online]. Available: https://blogs.technet.microsoft.com/mmpc/2016/05/26/link-lnk-to-ransom/. [Accessed 2 June 2016].