Scanning For Vulnerable Windows RDP Hosts

Enterprise Network Security

Dylan Bray
UT Austin

David Chun
UT Austin

ABSTRACT

We aim to detect systems vulnerable to CVE-2019-0708, also known as "BlueKeep," a critical vulnerability in the Windows Remote Desktop Protocol. We approach the problem by scanning and fingerprinting OS version, looking for an exposed RDP port, and then confirming vulnerability through a request/response comparison. We can aggregate information about vulnerable hosts to analyze the scope and severity of this vulnerability. We also compare our scan to existing tools such as Shodan.

1. INTRODUCTION

The proliferation of working remotely has led to the need for remote desktop access. Microsoft created the Remote Desktop Protocol (RDP) as a remote desktop solution for Windows. Millions of machines now have RDP enabled, even when it is infrequently or never used. In addition, many of these machines are exposed directly to the internet, rather than being secured behind a VPN, vulnerable to RDP attacks.

CVE-2019-0708 "BlueKeep" was first reported in May 2019 and affects all unpatched versions of Windows between Windows 2000 and Windows Server 2008 R2/Windows 7. BlueKeep allows arbitrary code execution through a bug that is exploited with a specially crafted packet. Windows RDP has services bound to any of 32 channels, and each SVC is created at startup and torn down on shutdown. However, attackers can request a SVC named MS_T120 and bind to any channel other than 31, which will cause a heap corruption and allow for arbitrary code execution [1].

To detect BlueKeep, we plan to scan the internet and fingerprint each operating system to find potentially vulnerable systems. We will then determine whether RDP is enabled by searching for its default port. With this detection, we will be able to compare vulnerable machines and analyze the current scope of the vulnerability. We will then be a able to compare these results with existing tools such as Shodan.

The first step is to complete a review of existing literature and determine the set up for creating our scan. We will then run the scan and collect relevant information for comparison with other tools. Based on this, we may modify or refine our scan and the information we collect to ensure more complete results. We will then report the commonalities in machines that we find are vulnerable.

2. MOTIVATION

At the base level, vulnerability scanners such as the one we address here exist to simply identify systems which are vulnerable to known CVEs and exploits, rather than actively identify a weakness or exploit a flaw. They are proactive in nature and give companies, individuals, and IT teams a practical tool with quantifiable outputs to better protect their own interests, data, and privacy. They pinpoint key areas of fixing, ensure successful patching, and perhaps more importantly encourage triaging of high priority threats. Additionally, many regulations and oversight committees mandate levels of security on their systems and vulnerability scanners can ensure adherence to proper standards.

Along these lines, a paper by Li, Avellino, Janies, and Collins (2016) [4] discusses a tool designed to improve enterprise vulnerability management via detection and monitoring called the Software Asset Analyzer (SAA). This scanner addresses a few key scenarios that could lead to vulnerable systems, namely unapproved software download, failure to upgrade, lack of knowledge of installed applications, and remediation verification. To detect the potential install of vulnerable CPEs and verify it has been properly dealt with, the SAA tool takes a blacklist and checks network hosts for unauthorized CPEs by analyzing their individual configurations. By first scanning the system, the SAA tool will mark it as safe if three distinct checks are passed: all required CPEs are present, no blacklisted CPE is present, and only one version of a CPE is installed at a time. Once marked as safe, this configuration is used for future comparison analysis, thus accounting for deviations between CPE configurations on different machines. While this particular tool concentrates on categorizing the CPEs, a next step mentioned in the paper is to "incorporate a scoring system such as CVE, NVD, and CVSS" [4] to assess overall vulnerability of network machines.

We will take this emphasis on analyzing for CVEs from this paper and scan for such vulnerabilities, specifically concentrating on the BlueKeep vulnerability. Developing such a network scanner enables that level of protection and oversight required to provide a secure network environment.

3. OUR ARCHITECTURE

Our tool first finds open network IP addresses with port 3389 open as this is required for RDP. It then further filters these discovered IP addresses looking for vulnerable IPs by fingerprinting each OS, first by http header and second via methods including a TCP/IP/SYN method described by Han and Du (2010) [3]. Once complete, the final step is to scan the filtered candidate IP addresses for the BlueKeep vulnerability. This scan can consist of detecting that RDP is running and then attempting a request/response comparison to check if the vulnerability has been patched or not.

The patch scan works by checking the response to a specially crafted packet, which will differ if the host is safe or vulnerable. The scanner first authenticates with the host via a handshake protocol and connects to the MS_T120 channel. Once we send a close channel command with a specified size, there are two possible outcomes. On a vulnerable machine, the host will close the channel properly, sending a disconnect packet. However, on a patched machine, the channel will not close and no disconnect packet is sent. Thus by waiting for a couple seconds, we can determine if the host has been patched or not [2].

Once we detect a BlueKeep vulnerable machine, we then proceed to extract holistic data about our targets, including OS version, other open ports, geographic location, among others using Whois and other network tools. This data is then used to aggregate common environment or network features which could give insight into the applicability, danger, and relevancy of BlueKeep, including geographic location and OS version of the host machine.

The actual test setup for scanning BlueKeep vulnerable machines includes 5 steps. First, we use zmap to constrain our IP search range to addresses which are responsive and have port 3389 open. We then take these discovered IPs and use scapy to send a SYN packet to a set of IP addresses, double checking for an open port 3389 to use for fingerprinting. Once we have filtered down to only a subset of the original IPs, we then use the SYN packet and the SYN/ACK reply to fingerprint the OS at the host IP address. While there are several ways of doing so, we are utilizing a passive OS fingerprinting method via scapy's p0f methods rather than an active implementation both for better accuracy as well as for simplicity. The key metrics are IP initial TTL and particularly TCP window size as they both can vary widely by OS and thus are good candidates for filtering. For example, Linux commonly has a TTL of 64 and TCP window size of 5840 while Windows XP usually has values of 128 and 65535 respectively. Windows Server/Vista/7 also vary, having TTL of 128 and TCP window size of 8192. Finally, once these candidate IPs have been identified, we will run the patch scan described above to detect whether they are truly vulnerable to the BlueKeep CVE. Those that are vulnerable proceed to our last stage where we collect data using Whois.

4. EXPERIMENTAL RESULTS

Our initial trial run of the fingerprinting and candidate selection step proved to match expectations, with 100% correctly determined IP results compared to what Shodan reports as those potentially vulnerable to BlueKeep out of a sample test set of 25 IP addresses. Out of 50, we fingerprinted 93% of the OS versions the same when compared to Shodan however, 1 of the conflicting IP addresses did not have port 3389 open at time of scanning. As previously explained, the IPs with open port 3389 and with a Windows OS Windows XP to Windows 7 are selected as our candidates for further exploration out of a larger set of IP addresses. We do this by sending IP/SYN packets via Scapy to this set of discovered IP addresses and fingerprint them with Scapy's p0f tool. The IP addresses that pass our filters can then be piped into our patch scanner to determine if they are running RDP and if so, is the machine properly patched.

However, not all potentially vulnerable machines are actually vulnerable. In our initial evaluation, the patch scanner iterated over a small list of candidate IPs and classified them into one of three categories: Vulnerable, Patched, and Offline. Vulnerable machines are shown to exhibit the properties that would allow for an attacker to successfully intrude. Patched machines are classified as all machines that respond to initial connection, but do not respond to the specially crafted packet, or force the connection closed. This behavior is likely due to firewall rules or other network-level measures intended to stop network scanning or exploitation of this specific type. Offline machines are simply those that do not respond to any packets. They may have been online during the fingerprinting phase but went offline since. About half of our candidate IPs were actually vulnerable, while the other half were patched. None were offline.

For the chosen BlueKeep vulnerability, we expect to see a more homogeneous group machines as the CVE applicable only for older OS without a patch. At a count of almost 500 thousand Windows systems still vulnerable as reported by Shodan, a number still half of the original release, we expect to find comparable ratios of vulnerable machines in our network scans, especially as BlueKeep was announced in Q1 of 2019. We also expect the detected machines to be running older websites, deprecated web stacks, and similar such dated endpoints. Additionally, these machines should be more likely to have high numbers of other possible vulnerabilities due to update neglect indicated by the lack of a BlueKeep patch. Since Windows 7 SP1 is the latest version affected by BlueKeep, we also expect to see vulnerable machines congregated in countries with poorer internet infrastructure as they will be those more likely to run a host OS released before 2012. When we gather this further data, we will create graphs comparing total IPs to candidate IPs, vulnerable to patched host IPs, Shodan results compared to our experimental data, as well as country comparisons, among others in order to properly and holistically evaluate how pervasive BlueKeep actually is within the Windows ecosystem.

With such a small initial results sample, we are unable to draw conclusions about the nature of vulnerable machines. However, we are able to get relevant IP information from Whois in addition to OS information. As we gather a larger results set, we will look for patterns among the machine classifications that may prove or disprove the previously mentioned hypothesis regarding where we expect to find BlueKeep. Further, we could compare these results to existing vulnerability databases to examine what vulnerabilities are co-resident with BlueKeep.

5. RELATED WORK

Since BlueKeep only targets Windows machines ranging

from Windows Server 2008 to Windows 7, we need to properly fingerprint each IP's host machine to determine likely candidates for this vulnerability. However, the header fields in the http request/responses may not be accurate. To increase confidence and properly identify the host machine OS, we can use an approach described by Han and Du (2010) [3] which identifies OS by TCP/IP fingerprint. This paper uses a method in which a host can be fingerprinted via an open TCP port. The method described in this approach constructs a TCP packet, initiates a connection handshake protocol, and records the protocol fingerprints found in the TCP response header. Once recorded, the data gathered is used to compare the behavior with a database of OS fingerprints to accurately judge the host OS type and version. Similarly, the paper by Shamsi, Nandwani, Leonard, and Loguinov (2014) [6] also describes a method of properly fingerprinting networks, but this time in the context of improving accuracy of the packet pipeline in the presence of non-ideal network conditions as we would encounter. There are quite a few limitations and mitigation steps revolving around this network latency as well as user modifications which affect the results however in general, the fingerprint method seems well-researched. These two options, when implemented properly, would greatly increase the certainty we would have when filtering out host OS compared to just using the headers.

Once the OS has been fingerprinted, the next step is to examine the system for the BlueKeep vulnerability. BlueKeep relies on RDP being installed on the host machine however, there are a few key indicators of vulnerability. The packet handling of RDP communications requires several modules, among them termdd.sys, a driver used to send mouse and keyboard actions. Upon further exploration, researchers discovered a channel named MS_T120 contained a vulnerability in which if a specially crafted packet is sent, the channel itself is freed. However, the pointer to the function is not freed and thus can be accessed on the channel, thus bypassing any prior authentication [5]. A scanner for BlueKeep can therefore bind this MS_T120 channel, send appropriate packets, close the channel, and compare the output to that of a patched machine. If they differ, then the host can be classified as being vulnerable to BlueKeep [2]. This approach for the scanner will be useful when searching for vulnerable machines out of those we have pre-filtered for OS and open port.

We also examined some works based around how BlueKeep exploits work and what they can do. In a paper released by Yan and Chen (2019) [8], they describe the impacts and risks of BlueKeep while also discussing the methodology of exploit potentially of use when building a network scanner to detect this CVE. They implement different methods of writing data into the host's kernel including via the Bitmap PDU, the Refresh PDU and the Client Name Request PDU, all abusing the "use-after-free" vulnerability previously described. The latter method is particularly useful as it enables an attacker to obtain a stable kernel pool which can be used to insert shell code. These risks can however be mitigated by the addition of traps while external scanning and exploitation of BlueKeep vulnerable hosts can be prevented and detected by the released patch. The exploits and methods discussed here give would be useful in the event we were not just scanning for vulnerable machines but also exploiting them.

Finally, we researched for papers discussing penetration testing and mitigation strategies on Windows Server in order to see how CVEs are detected and exploited on host machines in a Windows ecosystem as BlueKeep is a Windows only CVE. A paper by Stiawan, et al. [7] discusses how they found 12 different vulnerabilities present on Windows server and the penetration testing techniques they used. Exploits began by fingerprinting the OS and network via IP address/subnet mask information, OS and running service then progressing to techniques for elevation of privilege, ranging from guessing to cracking the target password. They leverage tools including Metasploit, Cain and Abel, as well as Netcat to obtain these privileges. Finally, they note future work in the area of extracting and classifying system data to properly identify new attack vectors as well as exploring methods of classifying threats to better protect the system from future attacks.

6. CONCLUSIONS

7. REFERENCES

[1] E. Caroll, A. Mundo, P. Laulheret, C. Beek, and S. Povolny. Rdp stands for "really do patch!" – understanding the wormable rdp vulnerability cve-2019-0708, 2019.

[2] S. Dillon. @zerosum0x0: reverse engineering, penetration testing, exploit development, May 2019.

[3] X. Han and X. Du. A new method about operating system identification. In *2010 2nd IEEE International Conference on Information and Financial Engineering*, pages 882–885, Sep. 2010.

[4] X. Li, P. Avellino, J. Janies, and M. P. Collins. Software asset analyzer: A system for detecting configuration anomalies. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 998–1003, Nov 2016.

[5] Z. S. Research. Bluekeep and forthcoming rdp attacks, 2019.

[6] Z. Shamsi, A. Nandwani, D. Leonard, and D. Loguinov. Hershel: Single-packet os fingerprinting. *ACM SIGMETRICS Performance Evaluation Review*, 42, 06 2014.

[7] D. Stiawan, M. Y. B. Idris, A. H. Abdullah, M. AlQurashi, and R. Budiarto. Penetration testing and mitigation of vulnerabilities windows server. *I. J. Network Security*, 18:501–513, 2016.

[8] T. Yan and J. Chen. Exploitation of windows cve-2019-0708 (bluekeep): Three ways to write data into the kernel with rdp pdu, Sep 2019.