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. BlueKeep is a “use-after-free” exploit and allows for arbitrary code execution on a host machine over a network. We approach the problem by scanning and fingerprinting OS version, looking for an exposed RDP port 3389, and then confirming vulnerability through a request/response comparison. We aggregate information about vulnerable hosts to analyze the scope and severity of this vulnerability. We also compare our scan to existing scans from 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, making them highly vulnerable to RDP attacks.

CVE-2019-0708 “BlueKeep” [2] 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 scan the internet and fingerprint each operating system to find potentially vulnerable systems. We then determine whether RDP is enabled by searching for its default port. With this detection, we compare vulnerable machines and analyze the current scope of the vulnerability. We then compare these results to data from existing tools such as Shodan.

After completing a review of existing literature, we determined the set up for creating our scan, which is accomplished in two parts. We first scan the internet for a list of candidate IP addresses that satisfies the required connectivity, port, and OS criteria. We then run a more complete scan that determines whether or not each of these candidates is in fact vulnerable. We then collect relevant information for comparison with other tools, analyze the commonalities in machines that we find are vulnerable and compare our results to Shodan.

2. MOTIVATION

At a basic level, vulnerability scanners such as the one we describe exist to identify systems which are vulnerable to known CVEs and exploits, rather than actively identify a new 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 for improvement, guide 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 ete al. (2016) [5] discusses a tool designed to improve enterprise vulnerability management via detection and monitoring called the Software Asset Analyzer (SAA). This scanner addresses some key scenarios that lead to vulnerable systems, namely unapproved software download, failure to upgrade, lack of knowledge of installed applications, and remediation verification. To detect the potential installation of vulnerable CPEs and to verify they have been properly dealt with, the SAA tool takes a blacklist and checks network hosts for unauthorized CPEs. 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” [5] to assess overall vulnerability of network machines.

We 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 SCAN

Because BlueKeep is a vulnerability with Windows RDP, our tool first finds network IP addresses with port 3389 - the default for RDP - open. It then categorizes these discovered IP addresses by fingerprinting each OS, first by http header and second via methods including a TCP/IP/SYN method described by Han and Du (2010) [4]. Next, it filters this list to create a list of potentially vulnerable candidates based on known vulnerable configurations as detected in the categorization phase. 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 [3].

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.

4. SCAN SETUP

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 previously 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. Finally, we use Shodan's API to compare the results of our scan to theirs.

5. 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 from XP to 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 fingerprinting them with Scapy's p0f tool. The IP addresses that pass our filters are then piped into our patch scanner to determine if they are running RDP and if so, is the machine properly patched.

With nearly 400 thousand Windows systems still vulnerable as reported by Shodan, only half as many as at the original release, we expected to find many vulnerable machines in our network scans. Once we validated our candidate IP selection toolchain would produce appropriate results, we ran it on a set of 74 million IP addresses as produced by Zmap. Out of these 74 million IPs, only 7.4 thousand of them had an open port 3389 and fewer still, around 1.4 thousand, were fingerprinted as Windows hosts. Out of the rejected IP addresses with port 3389 open, while the majority of them had no decipherable fingerprint, of the ones successfully identified, Linux with kernel 3.x was most common followed by Mac OS X. HP-UX also showed up in the fingerprinted results. Based on the ratio of addresses we scanned (74 million) to total public IP addresses (3 billion) with the number of IPs identified as vulnerable by Shodan (400,000), we would expect around 8,000 vulnerable IPs. We did not achieve this but we are in the same ballpark, leading us to trust that our scan is doing something useful.

From our full candidate IP set, 87% (1185) were classified as Windows Server 2008, Vista, or 7. The remainder were composed of Windows Server 2003 and XP. Since Windows 7 SP1 is the latest version affected by BlueKeep, we also expected 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. Gathering this data, we compared 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.

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. In the trial run, about half of our candidate IPs were actually vulnerable, while the other half were patched. None were offline.

When run with the full set of candidate IP addresses, similar ratios were found. 43% of the candidates were vulnerable, 38% were patched, and the rest were offline at time of scan. In particular, Table 1 describes the categorization results of running the scan we describe. We compare these results to Shodan as follows. Of the 523 IPs we identified as not vulnerable, Table 2 shows that Shodan agreed with 92.5% of these categorizations. Of the 595 addresses we identified as Vulnerable, Table 3 shows that there was significant disagreement. We only agree with Shodan 23% of the time. Further work could involve exploring this discrepancy to determine whether it is an issue with our method or with Shodan's scanning methods.

Table 1: Our Scan Categorization

Not Vulnerable (Patched)	Vulnerable	Offline
523	595	216

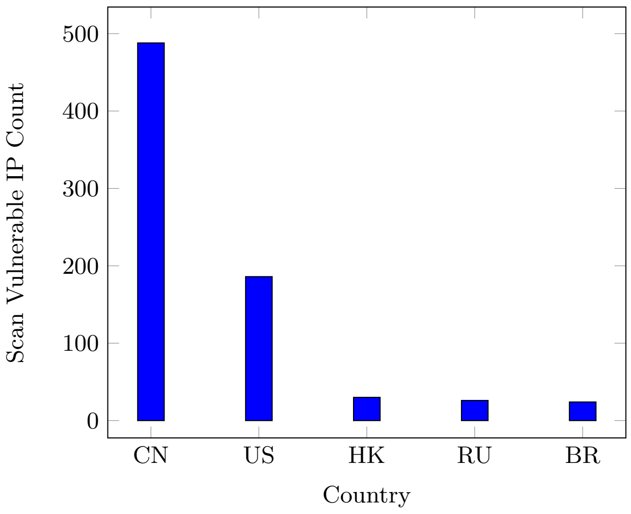
Table 2: Not Vulnerable

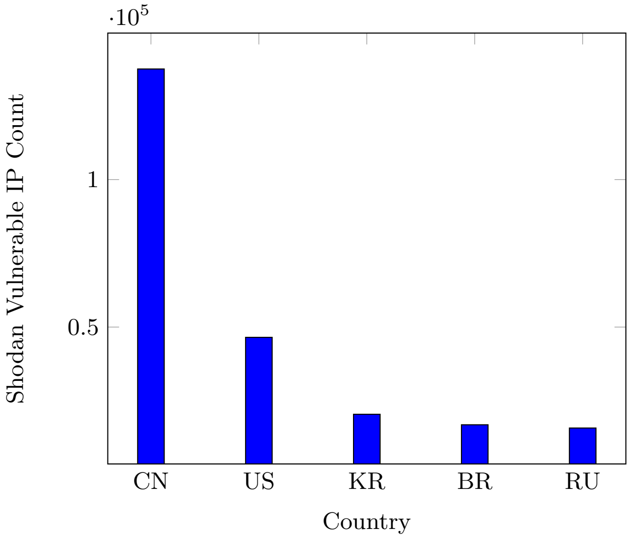
Both Not Vulnerable	Shodan Vulnerable
484	75

Table 3: Vulnerable

Both Vulnerable	Shodan Not Vulnerable
137	458

As shown in the following graphs, the ratio of vulnerable IPs by country is similar between our scan and Shodan. China by far has the most vulnerable IPs, followed by the US. This is despite the fact that China has a smaller internet infrastructure than that of the US, reinforcing the notion that China is less developed when it comes to security and technology upgrades. Unfortunately, Shodan doesn't have enough available data related to OS version to make a meaningful comparison to our scan's fingerprinting. However, Shodan did allow us to see what CVEs BlueKeep is often co-resident with. Unsurprisingly, many of these are related to vulnerabilities in Windows Server releases, mostly involving IIS and FTP services. We were also able to use Whois information to determine what platforms the vulnerable servers run on. We noticed that by far the most common were cloud services from Tencent and business software from Alisoft. Several Amazon EC2 and Microsoft Azure machines were also vulnerable. This leads us to believe that hosting providers and business software vendors could serve as a line of defense, scanning their own networks and the networks of their partners in greater detail, helping catch vulnerabilities.





6. 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) [4] 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) [7] 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 [6]. 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 [3]. 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) [9], 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. [8] 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.

7. CONCLUSIONS AND FUTURE WORK

Internet scanning is incredibly important for keeping a network secure and maintained. Since BlueKeep was exposed less than a year to date, it was an ideal CVE to test not only patch adoption but also to demonstrate the value of scanners such as ours in ensuring patches were applied correctly.

In our scanner, out of the initial set of 74 million IP addresses scanned, only roughly 1400 were qualified as candidate IPs, matching the correct OS and port requirements. Of this smaller subset, almost half of them were found to be still vulnerable to BlueKeep. Although a patch has been out for well over 6 months from time of writing, these machines either have not updated or the patch was not applied correctly. We then took these vulnerable hosts and used Whois to fetch metadata and generalize our findings to investigate for common environments.

Of these, our scanner identified about half as vulnerable, but these results don't agree completely with Shodan. We did see vulnerable hosts distributed geographically as expected, with by far the most in China. Many of these were running on cloud or business software providers, suggesting that these companies could play a role in looking for and notifying of vulnerabilities proactively. Future work in this are could involve investigating the best ways to accomplish this.

Future work along the lines of this BlueKeep scanner would include improving OS fingerprinting to increase the scope of IPs we check while still maintaining efficiency as well as extending this tool to other potential CVEs which would rely on similar port and service parameters. We would also explore the discrepancy between our results and Shodan's results.

8. 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] N. V. Database. Cve-2019-0708 detail, 2019.

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

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.