

Scanning For Vulnerable Windows RDP Hosts

Enterprise Network Security

Dylan Bray and David Chun

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 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

In a paper by Li, Avellino, Janies, and Collins (2016) [4], they discuss a tool designed to improve 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 black-listed 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 and scan for such vulnerabilities, specifically concentrating on the BlueKeep vulnerability for the Windows operating system. Once we filter out and detect the vulnerable machines, we can then collect network addresses, site type, and similar holistic data to gather statistics for BlueKeep vulnerable environments in order to try to detect commonalities.

3. OUR ARCHITECTURE

Our tool will scan the network for vulnerable IPs by fingerprinting each OS, firstly by http header and secondly via methods including a TCP/IP/SYN method described by Han and Du (2010) [3]. We can then proceed to filter by open ports and finally by scanning the actual host for the BlueKeep vulnerability. This scan can consist of detecting the RDP is installed and then attempting a request/response comparison to check if the vulnerability has been patched or not. Once we have detected a BlueKeep vulnerable machine, we can then proceed to extract holistic

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

data about our targets, including OS version, open ports, geographic location, among others. This data can then be used to aggregate common environment or network features which could give insight into the applicability, danger, and relevancy of BlueKeep.

4. EXPERIMENTAL RESULTS

For the chosen BlueKeep vulnerability, we expect to see a very wide range of vulnerable machines as the CVE is new and applicable only for older OS without a patch. At a count of almost 500k Windows systems still vulnerable as reported by Shodan, a number only half of the original release, we expect to find comparable ratios of vulnerable machines in our network scans. We also expect the detected machines to be running older websites, deprecated web stacks, and similar such dated software. 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.

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.

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, 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].

Finally, we examined some works based around how BlueKeep exploits work and what they can do. In a paper released by Yan and Chen (2019) [7], they describe the impacts and risks of BlueKeep while also discussing the methodology of exploit potentially of use when building a network scan-

ner 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.

6. CONCLUSIONS

7. REFERENCES

- [1] E. Carroll, A. Mundo, P. Lautheret, 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] 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.