Classifying RDP Remote Attacks on User Interfaces to Industrial Control Systems

Ryan P. Ramirez
Department of Computer Science
Naval Postgraduate School
Monterey, CA, USA
ryanreramirez@gmail.com

Thuy D. Nguyen
Department of Computer Science
Naval Postgraduate School
Monterey, CA, USA
tdnguyen@nps.edu

Neil C. Rowe
Department of Computer Science
Naval Postgraduate School
Monterey, CA, USA
ncrowe@nps.edu

Joseph T. Meier
Department of Computer Science
Naval Postgraduate School
Monterey, CA, USA
jmeier1291@gmail.com

***Abstract*—The Microsoft Remote Desktop Protocol (RDP) is popular for remote access, but its use for industrial control systems (ICSs) is risky because of their many vulnerabilities. Recognizing RDP attacks is also difficult because most RDP traffic is encrypted, and ICS traffic has many differences from non-ICS traffic. Our experiments obtained data from a hardened power-grid honeypot to characterize real RDP attacks on ICSs by malicious signatures, Windows event logs, and traffic metadata. Severity of malicious traffic varied widely and require novel labeling methods. This work can provide early warning to defenders about RDP attacks against ICS systems.**

Keywords—industrial control systems, remote desktop access, RDP protocol, honeypots, attack classification

I. Introduction

Industrial control systems (ICSs) have been slow to fix their software vulnerabilities because of the difficulty of updating their many kinds of specialized software, as well as their frequent need to remain continuously operational [1] [2]. At the same time, strong economic incentives encourage remote management of ICS systems using remote-desktop software [3]. Such software lets a user see the screen of a remote machine and interact with the screen as if they were seated in front of it, to do routine work or fix problems. Remote Desktop, TeamViewer, RemotePC, and Zoho Assist are examples. This could be especially valuable for ICSs like power grids that require periodic checking over wide geographic areas. Furthermore, many tasks that operators do on ICSs like checking dials and readouts, and changing switches and knobs, are easy to support with remote-access software [4].

However, remote-desktop software is also a popular target for cyberattacks on many systems because it gives a user considerable control of the system, enabling tampering with ongoing processes [5] and easy lateral movement within local-area networks [6]. Despite these threats, adoption of remote access to ICSs is growing due to the large potential benefits in efficiency. Some attacks already have used remote access, such as the attack in February 2021 on a Florida water treatment facility that exploited TeamViewer software to add dangerous chemicals to the water [7]. It is thus critical in defending ICS systems to develop better tools for identifying malicious activity in remote access.

In particular, we need tools to analyze traffic of the Microsoft Remote Desktop Protocol (RDP), which is popular and increasingly chosen for ICS access. RDP is a proprietary protocol developed by Microsoft to be used with its Remote Desktop Services and compatible clients. Given the poor track record of Microsoft bug fixes [8], using RDP for ICS system could cause many vulnerabilities [9]. Recent attacks such as those against Ukraine's power grid in 2016 [10] and 2022 [11], and the U.S. Colonial Pipeline attack in May 2021 [12], also confirm that ICSs are high-value targets for military and criminal operations. Weaknesses of ICSs also make them harder to defend: They typically use older software that no longer receives security updates, lacks the processing power to run security software while upholding timing requirements, and uses insecure protocols for communication between controllers [13]. Furthermore, new vectors of attack are now being created with ICS functions being moved to cloud servers.

Honeypots, decoy systems that mimic the behavior of real systems, can collect and analyze attack data. With custom additions, honeypots can simulate ICSs and collect data on ICS-specific attacks to provide signatures of new threats [14]. For our research, we deployed a honeypot with a simulated ICS user interface, and analyzed the attack data to learn features for characterizing attacks using RDP.

II. Previous Work

A. ICS Honeypots

ICS honeypots simulate services monitoring and controlling an industrial process. They can be deployed in operational ICS networks to warn about unauthorized access, or as standalone systems to collect attack intelligence. ICS honeypots are more complex than honeypots that imitate non-physical systems like a Web server. Low-interaction ICS honeypots may simulate a service for communicating with a PLC or responding to queries of process information. High-interaction ICS honeypots have sophisticated simulations of physical devices and the human-machine interfaces to control and monitor them.

An early honeypot for ICS systems was the SCADA HoneyNet Project [15] which simulated industrial protocols

This work was supported by the Department of Energy with Idaho Natl. Labs. and the Natl. Science Found. with the Scholarship for Service Program..

979-8-3503-2028-2/22/$31.00 ©2022 IEEE
DOI 10.1109/CSCI58124.2022.00126

Modbus and IEC 104, and collected data from attacks on them. [16] built a high-interaction SCADA honeypot that simulated a water treatment plant and found that most attacks targeted the software of the SCADA system rather than control of the industrial processes.

[17] deployed a realistic honeypot to simulate a small industrial prototyping company complete with a website and fake employee information. Virtual networking computing tools were publicly exposed on a workstation in their honeypot. Two ransomware and a cryptocurrency-mining attack were observed. Another project ran honeypots around the world on different cloud platforms [18]. On each platform, several honeypots were deployed including VNClowpot, a low-interaction honeypot that imitates a Virtual Network Computing service, and RDPY, a Python implementation of the RDP protocol. Their results showed that the most common attacks on cloud-service providers were against desktop sharing services.

B. The RDP Protocol

This work reported here focuses on the Microsoft Remote Desktop Protocol (RDP) [19]. Connections between remote-desktop client software (such as Microsoft Remote Desktop Connection) and a Remote Desktop Services host occur through TCP port 3389. RDP uses the T.120 suite of multimedia conferencing protocols. It uses multiple virtual channels for data transmission of the server's display data to the client, as well as the client's keyboard and mouse data to the server using the RC4 stream cipher.

RDP is a popular target of attack because of its insecure design. Brute-force attacks can be effective against systems with weak RDP credentials [20]. Also, vulnerabilities in Microsoft Remote Desktop Services and the RDP protocol itself have recently enabled BlueKeep and DejaBlue remote-code execution exploits which affect several versions of Windows running Remote Desktop Services [21]. Since these exploits can spread without human aid, they are often associated with EternalBlue, a Windows remote code-execution exploit which attacks the SMBv1 protocol. It was used for the WannaCry and NotPetya ransomware in 2017, causing billions of dollars' worth of damage worldwide. Even with Microsoft releasing patches for BlueKeep and DejaBlue, and with the National Security Agency issuing an advisory on BlueKeep exploitation, millions of unpatched machines likely remain vulnerable to these RDP-based attacks. A 2019 study quantified the global threat to RDP [22]. Researchers discovered different levels of persistence in attacks on remote-desktop tools, with some attackers trying to log in just a few times before giving up, and others spending days trying to figure out the machine's administrator credentials.

III. Our ICS Previous Honeypots

This research used honeypots that our group built previously using the Conpot and GridPot honeypots together on a platform provided by the DigitalOcean cloud service [23][24]. Conpot simulates services for the IEC 60870-5-104 (called "IEC104" for short), HTTP, Modbus, and S7Comm protocols to simulate a low-interaction ICS. GridPot is a backend for Conpot that simulates the components of an electrical grid, providing concrete assets for ICS attackers to attack. This means that in the honeypot we deployed, attackers can interact beyond a login screen. An earlier version of our honeypot improved GridPot and added a graphical user interface [23]. Subsequent work showed that cloud deployment and using the multi-honeypot deployment of T-Pot to implement it did not affect traffic [24]. However, we observed problems securing log data as several attackers successfully deleted it [25].

To better protect our honeypot, we hardened the logging mechanism in the GridPot implementation by separating out a user-interface machine, a logging machine, and a GridPot-simulation machine. The GridPot machine hosted ICS functions which could be accessed from a SCADA application on the user-interface machine, and the user-interface machine could be accessed remotely through RDP. Event logs and network traffic to GridPot and the user interface were sent to the logging machine to be securely stored.

Results from our experiments showed that the hardened architecture was mostly effective at securely storing attack data, though two attacks evaded logging by tampering with user-interface machine. Activity on December 26, 2021 executed a program Advanced_IP_Scanner.exe, but no other events indicated what it did or how it got on the machine. No RDP traffic was recorded during this time which might mean an attacker logged on at an even earlier time and dropped a program or script to execute the Advanced IP Scanner tool. This attacker likely evaded our logging system by either being careful not to trigger certain events or by erasing logs.

Later data starting on March 22, 2022 revealed many logins from IP addresses that claimed to originate from Ukraine, likely related to the conflict between Ukraine and Russia then happening. Processor use on the user-interface machine suddenly increased then from 30% to 53% at about the time of the last login to the Windows virtual machine. No user activity in the event logs explained increased processor use. On April 11, 2022 the Windows virtual machine crashed and reverted to the snapshot taken March 15 before putting it online. An attacker may have logged in to the Windows virtual machine on April 1, disabled logging, installed malware for processor-intensive tasks such as cryptocurrency mining (as we did find cryptocurrency executables), and disabled remote access to everyone but themselves. Alternatively, the crash may be due to a well-publicized attack called Industroyer2 [11] by the Sandworm Russian hacker group. The attack targeted Windows machines in an ICS network owned by a Ukrainian electric-grid operator, and could have spread to ICS networks outside of Ukraine. Attackers may have reverted our Windows virtual machine back to an earlier snapshot to conceal their activity. A scan from the Advanced Port Scanner tool suggests that the attacker determined our Windows machine was running in VirtualBox, since the IP addresses they scanned were default settings for it.

IV. Experimental Design

Although the only account accessible by the RDP protocol on our honeypot had minimal privileges, the attacker could still exploit unpatched vulnerabilities or misconfiguration to gain administrative privileges. Our solution (Fig. 1) was to run Windows in a virtual machine and to put Wireshark packet capture on a Linux host machine outside the Windows environment. Wireshark and logging could not be stopped by

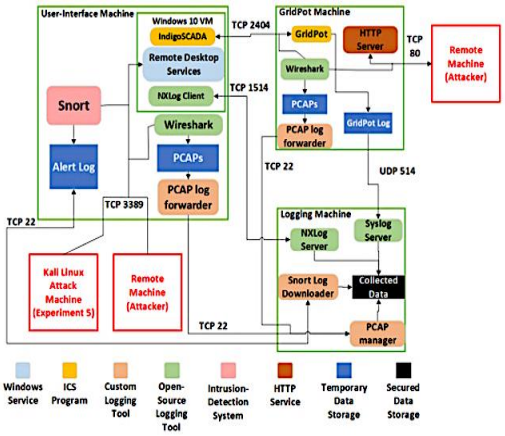

Figure. 1: Improved honeypot architecture with harden logging.

malware on the Windows system unless the attacker exploited a vulnerability in the hypervisor software to escape the virtual machine. Each virtual machine had a public network interface and a private interface to communicate between virtual machines.

To create instrumented datasets, we used open-source port scanners, vulnerability scanners, and Metasploit Framework exploits. We used port-scanning tools to generate benign routine scanning. Vulnerability-assessment tools, which analyze hosts and their running services, allow attackers to determine the best ways to attack and infiltrate networks. We ran these tools to generate data for the early stages of a malicious attack, though they could also be used for legitimate security monitoring. Metasploit was used to generate additional malicious data.

To identify features of attacks on RDP, we used recognizable attack patterns from the decrypted packet contents sent during the RDP connection, characteristics of the remote-desktop traffic, and attack behavior observed in video replays of remote-desktop sessions or assembled from Windows event logs. The finding of attack patterns used both the actual RDP traffic collected by our user interface, and data generated by port scanners, vulnerability assessment tools, and Metasploit modules. We used tools designed for RDP analysis to determine signatures of attacks, where they occur in the RDP connection sequence, and the severity of the attacks. Methods for attack characterization include comparing actual data to generated data, correlating RDP server event logs to captured data, and detailed analysis of RDP connection metadata.

Five experiments were done. Experiment 1 used PyRDP man-in-the-middle tool (www.gosecure.net/blog/2020/10/20/announcing-pyrdp-1-0/) to generate video replays of attacks and characterized attacks using these replays with captured network traffic and event logs. Experiment 2 let attackers interact with the remote-desktop service of the honeypot while using intrusion detection, network-traffic analysis, and event logs to catch exploit attempts. The user-interface machine was given a new name that was the name of a fake company, and given new user accounts. We used the entire Snort ruleset to detect signatures of attacks on other services, not just attacks on RDP, to see if attacks were trying to bypass the firewall to the user-interface machine. We also enabled the HTTP server of the GridPot machine, which displayed a single webpage containing brief information about the simulated power-grid and giving its public IP address to entice visitors. The logging machine logged Windows event logs, packets from the user-interface and GridPot machines, and HTTP requests for the new webpage. The logging machine used SSH to retrieve the Snort alert log and Web accesses every two hours.

Experiment 3 was identical to Experiment 2 except it used a different public IP address to see if a new address was more interesting to attackers. Experiment 3 received considerable traffic, so Experiment 4 repeated Experiment 1 with the PyRDP man-in-the-middle tool on the new address of Experiment 3. But Experiment 4 received much less traffic than that of Experiment 3, and we stopped Experiment 4 early. Experiment 5 just sent malicious packets directly to our honeypot to see what clues they provided. To generate these packets, we used tools from Metasploit, Greenbone, and Nessus, and also created custom malicious remote-desktop sessions to the honeypot. To simulate benign scanning traffic, we used the Nmap, Netcat, Angry IP, and Unicornscan scanners.

V. Results

A. Overall Results

Legitimate port scanning generated little RDP network traffic, so having few packets per session was a strong clue to legitimacy. Nmap generated the most traffic when configured to detect the RDP version: 78 packets, with only the first phase of the RDP connection sequence being completed. Nmap in its non-versioning configuration and Angry IP Scanner generated only four packets per session. Some signatures like the string "nmap" in the unencrypted data signaled nonmalicious scanners. The malicious traffic we tested generated more packets per session: 2061 for Nessus, 981 for OpenVAS, 272 for a Metasploit BlueKeep exploit, 205 for a Metasploit ForceExploit exploit, and 96 for Rdpscan. Only the Metasploit ms12_020_maxchannelids exploit had fewer packets than Nmap, 21 total. Two useful signatures, "nessus" and "openvas", were found in the RDP cookie field of the Client X.224 Connection Request packet that is sent in the first phase of RDP connection. Other signatures in cookies were "nbin" for Nessus and "openvasvt" for OpenVAS.

A payload signature for the Metasploit ms12_020_maxchannelids exploit was also found in cleartext. A malicious value is visible in the maxChannelIDs field, intended to cause denial of service. Signatures for attempted BlueKeep attacks or vulnerability scans could be observed in the unencrypted settings-exchange phase of the RDP connection sequence. Random 7-character or 8-character RDP cookies were signatures of BlueKeep exploitation, vulnerability scans using Metasploit modules, and the Rdpscan tool.

Windows event logs also indicated different kinds of RDP traffic. Event ID 261 ("listener for RDP received a connection") was the only event associated with benign traffic. Malicious tools using RDP triggered Event IDs 261, 1158 ("remote desktop services accepted a connection"), and 40 ("remote-desktop session disconnected") since they proceeded beyond the

first phase of the RDP connection sequence. We also saw event ID 1149 ("user authentication succeeded for remote desktop services") with our attack tools in Experiment 5.

To roughly categorize RDP attack severity, we used five levels: very low (the attacker has only partially completed the connection sequence), low (the attacker connected but spent less than a minute), moderate (the attacker connected for 1-5 minutes), high (the attacker connected and either spent more 1-5 minutes or more than 5 minutes with no log entries), and very high (the attacker spent more the figure minutes connected and generated log entries). We also separately classified RDP connections into the categories of clearly malicious, very suspicious, suspicious or benign. Clearly malicious connections included those with signatures of known malicious attacks such as the Metasploit ms12_020_maxchannelids signature and the RDP cookie signatures that we got from traffic for Nessus, OpenVAS, Rdpscan, and Metasploit BlueKeep. All traffic from a source labeled as clearly malicious was also labeled that. Lacking a signature, a connection was classified as very suspicious if more than 500 packets were exchanged between the client and server, since this suggested that RDP initialization was completed and a desktop session was established. All traffic from a source labeled as very suspicious was also so labeled regardless of size. A connection was classified as suspicious if an exchange of 85 to 500 packets was observed during an RDP connection, as this indicates the partial completion of the RDP connection sequence. RDP connections less than 85 packets and from unmalicious sources were classified as benign based on our analysis of the traffic of the nonmalicious tools. Table 1 shows overall statistics on the experiments.

B. Experiment 1

In Experiment 1, we confirmed establishment of remote-desktop sessions through Windows event logs for the four connections with exchanges greater than 500 packets, though each session was disconnected quickly. About 90% (122,180) of the RDP connections used the cookie "hello"; second most common was "administr", which is reported associated with Russian crawlers [26].

After applying all methods of detecting malicious RDP network traffic through signature detection and RDP connection metadata, we characterized 1071 RDP connections as malicious, and 133733 connections as benign with very low severity because none came from sources that sent malicious traffic. PyRDP also automatically logged BlueKeep events. Of the 21 randomly generated eight-character alphanumeric RDP cookies that we observed, 17 related to BlueKeep scanning attempts. For "very suspicious" sessions, the video replays generated by PyRDP showed that no attackers made it beyond the Windows login screen. From this and lack of attempts to scan the network or exploit BlueKeep recorded in the event logs, we concluded that the severity of these attacks was low.

C. Experiment 2

In Experiment 2, the top ten RDP cookies used during RDP connections were similar to those of Experiment 1 except for 867 occurrences of the name of our Windows machine, of which 853 came from the Netherlands. All the connections in this experiment that were characterized as clearly malicious attempted BlueKeep scans against our honeypot. We decrypted the RDP application packets sent during these connections to search for signatures of BlueKeep. All connections that used a randomly generated RDP cookie also tried to create the MS_T120 virtual channel, which also indicated BlueKeep scanning or exploitation. This confirmed that random strings in cookies are a good clue to BlueKeep. Malicious behavior was not observed in the two RDP connections characterized as suspicious from 45.132.226.221. One was 476 packets long and did not complete the whole connection sequence. The other RDP connection (from the same source) was 20 packets long and occurred one second before; it only completed the first phase of the RDP connection sequence before ending the connection. This was likely a port scan to determine if the RDP service was running before performing a deeper probe. All remaining connections were each under 30 packets long and likely benign.

D. Experiment 3

Experiment 3 had considerably more traffic than any of the other experiments, despite having the same configuration as Experiment 2; apparently the change in Internet address made a big difference. The top RDP cookies were like those of Experiments 1 and 2. 71 RDP connections were classified as "clearly malicious" with low severity and were mostly BlueKeep exploits; we observed 11 RDP connections with signatures of Rdpscan, and their sources created 60 other RDP connections. 102 RDP connections were observed exchanging more than 500 packets with the Windows RDP service, and their malicious sources created 62990 other RDP connections. The 102 RDP connections with more than 500 packets established remote-desktop sessions. The sources that created these 102 RDP connections created 62990 other RDP connections to our user-interface machine throughout the experiment, most of which appeared to be RDP service probes less than 30 packets long. These 62990 RDP connections were classified as very suspicious, and each was given the same severity classification as its source's highest. To evaluate attacks classified as "suspicious", we decrypted their RDP packets to confirm that none had attempted a BlueKeep scan or exploit by creating the MS_T120 virtual channel. We did not observe any BlueKeep-related signatures or other malicious behavior in these "suspicious" packets. We decrypted the network traffic of the "clearly malicious" connections and found the MS_T120 signature of BlueKeep in each. For "very suspicious" traffic, we compared the events in the Windows event logs against artifacts left behind on the virtual machine. This included checking PowerShell transcripts, new files in the file system, and new keys in the registry.

Not every remote-desktop session left behind artifacts of attack, but we could classify the severity of attacks that did. From the attack on July 22 from a German source, two files "data.exe" and "c2.exe" were created in the Documents folder of the public user. Metadata for each file could be seen in the Sysmon event logs. The original file name of data.exe was reported as "VBCECompiler", and the original name of "c2.exe" was "DedicStore.SystemInfoChecker.exe". The MD5 hashes for each file were checked on VirusTotal [27], which confirmed their original names and reported that several security vendors had flagged the files as malicious. Artifacts for each malware file could be seen in the Windows registry.

TABLE 1: SUMMARY OF EXPERIMENTS. NUMBERS FOR EACH ENTRY REPRESENT COUNTS OF SEVERITY VALUES OF "VERY HIGH", "HIGH", "MODERATE", "LOW", AND "VERY LOW".

Experiment	Duration (days)	Total RDP connections	Clearly malicious	Very suspicious	Suspicious
1	24	134,804	0/0/0/1052/0	0/0/0/9/0	0/0/0/10/0
2	8	13,676	0/0/0/0/13	0/0/1/7/0	0/0/0/0/2
3	26	2,059,623	111/24/37882/25075	0/0/0/0/4878	0/0/0/0/4878
4	3	6,809	0/0/0/0/0	0/3/0/2/0	0/0/0/0/0

An attack from a Nigerian source on August 5, 2022 changed the file system by adding "laravel_scanner.exe", "SenderSMS" files and directories, and several executables such as "pytransform.dll", "python27.dll", and "msvcm90.dll", which were likely needed by the scanner. From event logs, the processes that created these made many DNS (domain-name) requests to Web servers including "tactyl-services.com", "cdlima.org.pe", "epu.edu.pe", "handle-wakemag.fr", and "chuckneedham.com". We could not connect to "tactyl-services.com", "cdlima.org.pe" appeared to belong to a Peruvian engineering college, "epu.edu.pe" redirected us to the University of San Martin de Porres in Peru, "handle-wakemag.fr" was the site for a French water sports magazine, and "chuckneedham.com" was a relatively simple WordPress blog. These appear to be Web sites compromised by the same attack. The 254-kilobyte "Results.txt" file on the Desktop listed HTTP Web servers. Inside the "SenderSMS" directory, two files called "SHELL.txt" and "RDP.txt" contained what appeared to be login credentials and an IP address. Also, folder "SenderSMS", not previously observed, contained "CHASE 2022.zip" which in turn contained a folder called "SpoxV5" with PHP scripts and other Web-related documents like "robots.txt", and "visit_log.txt".

On August 8, 2022 we noticed a spike in processor resource use. This related to an attack from Australia which installed the same cryptocurrency miner "xmrig.exe" also observed in [24]. We restored the Windows machine to its original state before it was attacked, but within ten minutes, another attack from Australia installed the same software in a different place. We classified these sessions as "very high severity" due to their changing the Windows machine.

E. *Experiment 4*

We restored our Windows machine to a clean state before running Experiment 4. This experiment used PyRDP MitM in the same configuration as Experiment 1 with the same public IP address as Experiment 3. Unlike the previous experiments, the most common RDP cookie was the public IP address of our user-interface machine. No indicators suggested that attackers had used Nessus, OpenVAS, or Metasploit ms12_020_maxchannelids. We also did not observe any randomly generated seven-to-eight-character RDP cookies that would indicate BlueKeep. No RDP connections were classified as "clearly malicious". Five RDP connections were classified as very suspicious; two RDP connections were observed exchanging more than 500 packets with our RDP server, and the three other RDP connections from their sources were classified as very suspicious by association.

F. *Comparing the Experiments*

Using PyRDP MitM to intercept and forward attacker connections to our honeypot in Experiments 1 and 4 dramatically discouraged attacks as seen in Table 1. Clearly, attackers notice that PyRDP handles RDP network traffic differently than direct connections and they do not like it.

In Experiment 3, changing the honeypot IP address definitely helped increase traffic. It appears to have received a boost from the previous use of the address by a Web server, as indicated by the previous scanning data from Shodan. Shodan reported open ports for SSH, HTTP, and HTTPS on the machine, of which only the second was true, and did not report port 3389 (the RDP port) as open until the 23rd day of the 26-day run. Traffic declined significantly over Experiment 3 (Fig. 2), apparently because it took a while for attackers to recognize the site had significantly changed. Probably our honeypot was attacked at a high initial rate because attackers thought a Web server was running. This does suggest that it is important to rotate addresses for honeypots.

We did not see any attacks using the Metasploit ms12_020_maxchannelids module to exploit our Windows machine. This is likely due to the age of the exploit and the number of Windows machines that are now patched against it. We observed many attempts to scan for and exploit BlueKeep, which is likely favored for attacking newer versions of Windows because it can access machines instead of only causing denial of service. While we had methods to characterize malicious vulnerability scanning using Nessus and OpenVAS, we did not observe any attackers using them to scan our honeypot. Scanning time could likely be the cause. In our testing, it took over 20 minutes to scan all common ports of our honeypot, and over 7 minutes to scan only port 3389. In our case, the attackers seemed to quickly realize that they could easily log into our Windows machine. Since our honeypot's login method was insecure by design, attackers had little reason to scan for vulnerabilities.

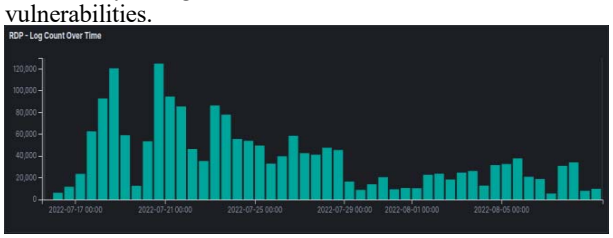

Figure 2: Traffic count over Experiment 3.

The Snort intrusion-detection system was ineffective at detecting attacks on RDP due to its inability to decrypt most traffic. Examining the duration of the session and artifacts left behind did help identify attacks instead.

VI. Conclusions

We conclude that attacks involving the RDP protocol can be recognized despite its use of encryption and the subsequent ineffectiveness of traditional packet-based intrusion detection. Live RDP attacks involved significantly longer sessions than normal scanning, left different messages in log records, and left cookies and downloads we could recognize as malicious. This may suffice for finding many attacks on ICSs using RDP, but proper hardening of the logging architecture is essential.

Acknowledgment

Opinions expressed are those of the authors and do not represent the U.S. Government.

References

[1] T. Morris, R. Vaughn, and E. Sitnikova, "Advances in the protection of critical infrastructure by improvement in industrial control system security," Proc. 11th Australasian Information Security conference, Adelaide, Australia, 2013.

[2] O. Idrissi, A. Mezrioui, and A. Belmekki, "Cyber security challenges and issues of industrial control systems – some security recommendations," Proc. 5th IEEE Smart Cities Conference, 2019.

[3] J. Garcia et al., "Reconfigurable distributed network control system for industrial plant automation," IEEE Transactions on Industrial Electronics, Vol. 51, No. 6, December 2004.

[4] M. Furuya, T. Fujkomot, and T. Sekozawa, "WWW-browser-based monitoring system for industrial plants," Proc. 25th Conf. of the IEEE Industrial Electronics Society, 1999.

[5] P. Ackerman, "Industrial cybersecurity," Birmingham, UK: Packt, 2017.

[6] T. Bai, H. Bian Abbas Daya, M. Salahuddin, N. Liman, and R. Boutaba, "A machine-learning approach for RDP-based lateral movement detection," Proc. 44th IEEE Conf. on Local Computer Networks, Osnabrueck, Germany, October 2019.

[7] Cybersecurity and Infrastructure Security Agency, "Compromise of U.S. water treatment facility," February 2021. https://www.cisa.gov/uscert/ncas/alerts/aa21-042a

[8] M. R. Lidestri and N. C. Rowe, "Quantifying the milestones of cyber vulnerabilities," Proc. of the 21st International Conference on Security and Management, Las Vegas, NV, US, July 2022.

[9] N. Danchencko and G. Mazurenko, "Detecting and analysis malicious activity on remote desktop protocols using integrated security system," Proc. Conf. of Russian Young Researchers in Electrical and Electronics Engineering, Moscow, 2018.

[10] Cybersecurity and Infrastructure Security Agency, "CrashOverride malware," June 2017. https://www.cisa.gov/uscert/ncas/alerts/TA17-163A

[11] ESET Research, "Industroyer2: Industroyer reloaded," WeLiveSecurity, April 2022. https://www.welivesecurity.com/2022/04/12/industroyer2-industroyer-reloaded/

[12] Office of Cybersecurity, Energy Security, and Emergency Response [OCESER] (n.d.), "Colonial pipeline cyber incident," August 11, 2022. https://www.energy.gov/ceser/colonial-pipeline-cyber-incident

[13] S. Mathezer, "*Introduction to ICS Security*," SANS, May 2021. https://www.sans.org/blog/introduction-to-ics-security/

[14] J., Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems," ArXiv:2108.02287 [Cs]. 2021. http://arxiv.org/abs/2108.02287

[15] V. Pothamsetty and M. Franz, "SCADA HoneyNet project: Building honeypots for industrial networks," 2004. http://scadahoneynet.sourceforge.net

[16] Ó. Navarro, S. Balbastre, and S. Beyer, "Gathering intelligence through realistic industrial control system honeypots. critical information infrastructures security," CRITIS, 2019. https://doi-org.libproxy.nps.edu/10.1007/978-3-030-05849-4_11

[17] S. Hilt et al., "Caught in the Act: Running a realistic factory honeypot to capture real threats," Trend Micro Research, 2020. https://documents.trendmicro.com/assets/white_papers/wp-caught-in-the-act-running-a-realistic-factory-honeypot-to-capture-real-threats.pdf

[18] C. Kelly, N. Pitropakis, A. Mylonas, S. McKeown, and W. J., Buchanan," A comparative analysis of honeypots on different cloud platforms," Sensors, 21(7), 2433, 2021. http://dx.doi.org/10.3390/s21072433

[19] Microsoft, "Remote desktop protocol," 2020. https://docs.microsoft.com/en-us/windows/win32/termserv/remote-desktop-protocol

[20] J. Buchanan, J., "Securing RDP vulnerabilities: learnings from Bluekeep and DejaBlue," Rapid7, November 2019. https://www.rapid7.com/blog/post/2019/11/07/the-anatomy-of-rdp-exploits-lessons-learned-from-bluekeep-and-dejablue/

[21] Cybersecurity and Infrastructure Security Agency, "NSA releases advisory on BlueKeep vulnerability," June 2019. https://www.cisa.gov/uscert/ncas/current-activity/2019/06/04/NSA-Releases-Advisory-BlueKeep-Vulnerability

[22] M. Boddy, B. Jones, and M. Stockley, "RDP exposed—the threat that's already at your door," Sophos, 2019. https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-rdp-exposed-the-threats-thats-already-at-your-door-wp.pdf

[23] N. Rowe, T. Nguyen, M. Kendrick, Z. Rucker, D. Hyun, and J. Brown, "Creating effective industrial-control-systems honeypots," Proc. Hawaii Intl. Conf. on Systems Sciences, Wailea, HI, January 2020.

[24] N. Rowe, T. Nguyen, J. Dougherty, M. Bieker, and D. Pilkington, "Identifying anomalous industrial-control-system network flow activity using cloud honeypots," Springer Lecture Notes, Proc. National Cyber Summit, Huntsville, Alabama, September 2021.

[25] J. Meier, T. Nguyen, and N. Rowe (2023), "Hardening honeypots for industrial control systems," Proc. Hawaii Intl. Conf. on Systems Sciences, Maui, HI, January 2023, in press.

[26] Tabdiukov, "Mstshash=administr explained," GitHub. https://github.com/olipo186/Git-Auto-Deploy/issues/221

[27] VirusTotal (n.d.)," VirusTotal—home," 2022. https://www.virustotal.com/gui/home/upload