Proposal of Tools for Intrusion Detection - Nexpose

Brian Fajardo
Department of Computer Science
Montclair State University
Montclair, United States
fajardob1@montclair.edu

Obinna Ezeadum
Department of Computer Science
Montclair State University
Montclair, United States
ezeadumo1@montclair.edu

Taghreed Alshammari
Department of Computer Science
Montclair State University
Montclair, United States
alshammarit1@montclair.edu

Abstract—Visibility in an IT infrastructure is a crucial remark that must be accounted for. When there is no visibility in an IT infrastructure, the entire infrastructure is at a much higher risk of suffering from vulnerabilities which can quickly lead to threats and damage the infrastructure even more. This leads to a question. How can you secure what you don't know what you have? This question is fundamental and will be the basis of our proposal. Without proper visibility you are essentially just protecting what you **think** you have. That is the difference there. Addressing visibility in an IT infrastructure is a great way in managing and remediating exposed vulnerabilities that malicious actors can take advantage of. One such approach is by using Intrusion Detection tools. Nexpose by Rapid7 is the tool that we will analyze and investigate whether it provides the visibility criteria that we are looking for. This criteria involves whether there is reporting features and vulnerability assessment that can ensure sufficient visibility where a System Administrator can ensure that the IT infrastructure is well protected.

I. INTRODUCTION

We wanted to first investigate how Nexpose can help ensure visibility in an IT infrastructure. According to Rapid7 Nexpose possesses features such as real risk score 1-10 CVSS which results in thousands of critical vulnerabilities. The vulnerability scanner's real risk score provides more actionable insight. This is a feature that will surely help a System Administrator in taking action whether to remediate findings of a vulnerability to help ensure no vulnerability is left exposed or unnoticed. In terms of visibility this satisfies our criteria of reporting and an assessment where a decision can be made whether to take action or to hold off in order to gather more information. Nexpose evcn has a built in scale from 1-1000 that highlights the vulnerabilities most likely to be used in an attack. This will help prioritize and in some ways remove pressure from decision making ensuring that truly critical issues are addressed first. Another feature that cannot be ignored is policy assessment. There is integrated policy scanning which can be used to benchmark systems against popular standards like NIST. Our criteria of reporting and vulnerability assessment will be greatly benefitted here. The National Institute of Standards and Technology is a highly reputable resource here because it will ensure compliance of any possible findings of vulnerabilities. This is also a great way in making sure that organizations are held accountable in handling their cybersecurity risks. In the real world organizations must be liable in some way when they fail to remediate vulnerabilities that have later caused a threat to escalate and cause damage.

II. LEVERAGING VISIBILITY FOR CVE

CVE stands for Common Vulnerabilities and Exposures. It is a list of entries each containing an identification number, a description, and at least one public reference for publicly known cybersecurity vulnerabilities. Vulnerabilities come in all shapes and sizes. This is helpful when a specific vulnerability wants to be assessed and managed. We need to take into consideration that many vulnerabilities will score very low on the Common Vulnerability Scoring System (CVSS). This scoring measure simply allows for assessing the severity of a system. If a CVE is given a high CVSS where the max is a 10, then immediate action can be taken because the CVSS score has determined that the system's severity is critical and remediation must be taken promptly. These types

of measures allow for proactive decisions to be made. For example, now that there is visibility in the criticality of a system from a CVE based on its CVSS, there is enough justification in remediating this specific CVE instead of one that scored a lower CVSS. The main point to understand here is leveraging a tool like Nexpose which will take a CVE template and find out whether any system in the IT infrastructure is vulnerable.

III. THE VULNERABILITIES PAGE

The vulnerabilities page in Nexpose will be helpful since it will cover extensive graphical information. We plan on using this page in our report. The key areas that we are looking for are filtering capabilities where vulnerabilities can be filtered based on their exposure. For example we want to cover any findings in exploitation which are vulnerabilities that are active in the wild and are publicly known. The difficult challenge we see ourselves encountering is whether there are protective measures against zero day vulnerabilities. We are looking to see whether Nexpose offers some kind of guidance or protective measures to ensure that an IT infrastructure does not suffer from zero day vulnerabilities which are unknown to the security community and are typically in the wild ready to be exploited at any given moment. Having a clear visibility on how a zero day vulnerability can be exploited is a great way in preparation for when the zero day vulnerability becomes publicly known. This will achieve both remediation and allow for organizations to decide on what plan of action to take on the recent vulnerability. Nexpose offers a feature called Exploit Exposure which can be leveraged to verify vulnerabilities to focus on remediation tasks on the most critical gaps in security. The benefit in using the Exploit Exposure feature is that it increases awareness in finding out whether for the discovered vulnerability there is an associated exploit. Something to keep in mind however is that although the required skill level for the exploit is shown, security cannot be overlooked regardless whether the exploit poses very little harm or whether the exploit skill level is rudimentary. Security should never be overlooked or taken lightly. It is crucial to have a zero trust model or strategic thinking when it comes to analyzing security.

IV. ZERO TRUST SECURITY MODEL

The Zero Trust security model is not just a fancy buzzword. It is an actual strategic initiative that helps prevent successful data breaches by eliminating the concept of trust from an organization's network architecture. The principle lies in the saying "never trust, always verify". This security model is like a reminder to never trust everything you see. There needs to be some kind of verification process that can prove whether something should be trusted or not. Cryptographic protocol schemes like digital signatures are a great relatable example of verification being conducted in the background. Similarly to ensure that vulnerabilities are being reported correctly, we will need to analyze the Nexpose reports from the scans conducted. An issue that can stem from scans are false positives and false negatives. Although we are analyzing how the zero trust security model can be used to remove trust from a network, we must still make sure that trust is a clear benchmark in our results. For example there needs to be trust in the results that are conducted from the scans. False positives can be dangerous when left unnoticed. These kinds of vulnerabilities must be accounted for and dealt with immediately. Similarly false negatives where a certain vulnerability is known to be actually good and can be something like a critical service for a company that was mistakenly classified as negative. Overall the main idea here is to follow the zero trust model in every security initiative but to not completely abandon trust. There still needs to be some trust that is justifiable and helpful for the network as a whole. A huge benefit in using the zero trust security model comes from using the zero trust architecture. In Zero Trust you identify a "protect surface". The protect surface is made up of the network's most critical and valuable data, assets, applications and services. They are unique to every organization. Since it contains only what's most critical to an organization's operations, the protect surface is orders of magnitude smaller than the attack surface and is always knowable. This protect surface is like adding a perimeter around the surface where everything inside the surface can be managed effortlessly without needing to struggle with finding out whether a device is inside or outside the protected surface. This will strengthen visibility considerably and is something we will address more in depth. With the protect surface identified, you can identify how traffic moves across the organization in relation to the protect surface. Added benefits in using the zero trust security model include understanding who the end users are like their behavior on a device, which applications they are using and how they are connecting is the only way to determine and enforce policy that ensures secure access to your data. There first needs to be a clear understanding of all of these interdependencies. Once there is a clear understanding then you should put controls in place as close to the protect surface as possible. The end result of doing this is ensuring visibility in the form of a microperimiter around the protect surface. The microperimeter will move in tandem with the protect surface resulting in minimal configuration downtime needed.

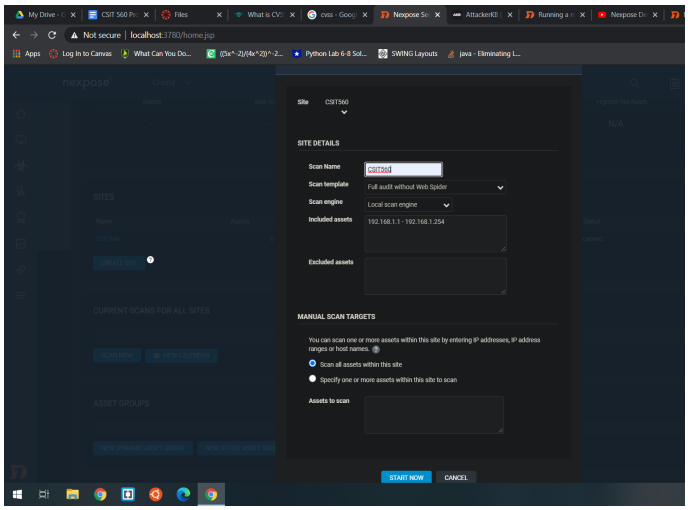
V. MILESTONES AND DATES

We plan on analyzing more features on Nexpose that can further address our problem, "How can you secure what you don't know what you have?". Further we will see if Nexpose is sufficient in addressing our problem. We will research more features this weekend. 6/6 - 6-/7.

VI. RUNNING OUR FIRST NEXPOSE SCAN

Before analyzing additional features on Nexpose, we decided it was best to first run a scan using the default local scan engine. To prepare for the scan we needed to first set up our scan environment. In order for a scan to be conducted there needs to be something that we can actually scan! This component is called a site. In the "What is a site?" help guide for Nexpose, a site is defined as a collection of assets "that are targeted for a scan." You must create a site in order to run a scan of your environment and find vulnerabilities. In the setup process we created a site called CSIT560. The scan template is used to define the scan criteria of how we want Nexpose to go about in investigating our targets. Full audit without a web spider is the default scan template. Since we are not scanning any web applications there is no need for a web spider. Scan engine is the workhorse behind the Nexpose solution that's performing the vulnerability assessment and then reporting the actual results back to your Nexpose security console. Last but not least we specified the IP address range of the test network being my home network. The IP address range is from 192.168.1.1 - 192.168.1.254. This will be the range where included assets will be scanned (scan all assets within this site).

SITE CSIT560 DETAILS

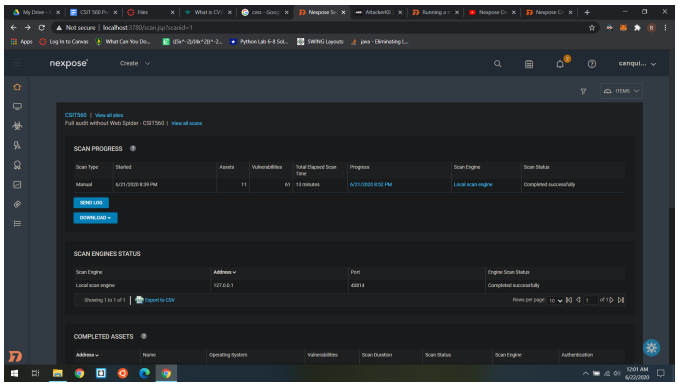


With our site CSIT560 created we were now ready to run our first scan!

VII. SCAN RESULTS

The scan of our site CSIT560 had some interesting results. It was able to scan 11 assets and discover 61 vulnerabilities. The total scan time was 13 minutes and it was completed successfully without any errors. In the Completed Assets section we can see the IP address of each respective asset that was identified in the scan. The name of the asset and the operating system of each asset followed by the number of vulnerabilities that was discovered for each asset. Lastly there were no incomplete assets or any asset that could not be completely scanned because they went offline.

SCAN PROGRESS/SUMMARY



The most interesting findings in our scan is the number of vulnerabilities that were found in the Fios_Quantum_Gateway.fios-router.home host. A total of 31 vulnerabilities were discovered for this host. This was the asset with the highest number of vulnerabilities discovered. There were several assets with zero vulnerabilities discovered and a few with only 4, 2, or 1. These two pieces of information can be classified as outliers. Something worth addressing however is that just because an asset has zero vulnerabilities does not mean this asset has no vulnerabilities. Since we ran a default vulnerability scan assessment, we did not change any of the configuration settings like implementing a policy scanning or a more aggressive style scanning called discovery scan. The purpose of this first scan was to see how robust the default local scan engine is when configured with its default settings and test it against our criteria whether or not there is sufficient visibility in the test infrastructure of our first scan. We looked to see if there were any reporting features like a chart to examine why certain vulnerabilities received a higher CVSS score or risk score. On the other hand a discovery scan could be very useful for us because Rapid7 describes it as more aggressive and "can produce more thorough and accurate results." This may discover vulnerabilities for those assets that were reported to have zero. With the information obtained from the scan, the next plan of action was to begin the assessment.

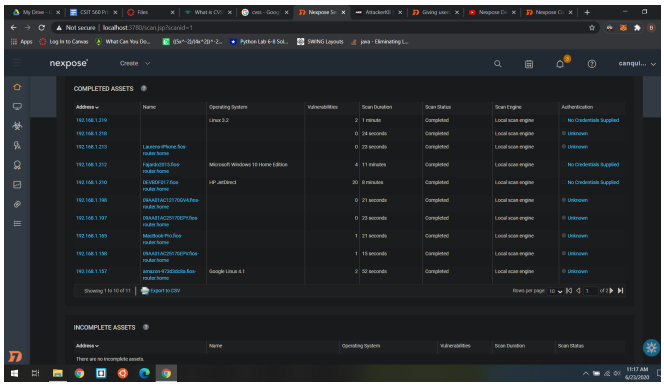
VIII. ASSESSMENT

The first target in the assessment was the asset that was discovered with the most vulnerabilities Fios_Quantum_Gateway.fios-router.home. With a total of 31

vulnerabilities discovered we felt that this was the right decision to make considering since this asset has been classified as an outlier for us. No other asset comes close to having this many vulnerabilities, the second most discovered vulnerabilities being found on my HP JetDirect printer with 20 vulnerabilities discovered. However with the 31 vulnerabilities that were discovered on my Fios home router, this amount can be very overwhelming. What would have happened if this number was significantly greater like 100? This is something that was on our mind when beginning the vulnerability assessment phase. Instead of looking at each and every vulnerability, we focused instead on the ones that scored a high CVSS and risk score. This was our priority first in tackling those vulnerabilities that pose a higher risk of being exploited and becoming an active threat to our test infrastructure.

COMPLETED ASSETS PAGE

- 11 Assets were discovered. Last asset that was discovered with 0 vulnerabilities is not shown here.
- Plan of action was to first assess the Fios_Quantum_Gateway.fios-router.home asset. This asset has been discovered with 31 vulnerabilities the most out of all the 11 assets that were scanned.

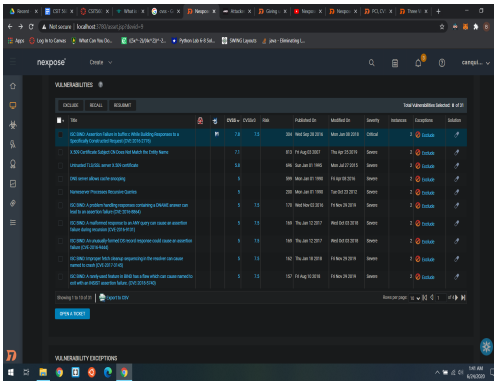


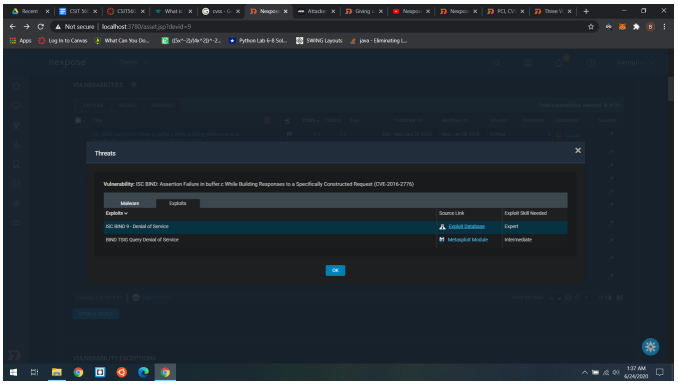
In the Vulnerabilities section of the Assets page for Fios_Quantum_Gateway.fios-router.home, we are greeted with the vulnerability that scored the highest CVSS score 7.8 with a severity level of critical. The vulnerabilities are ordered from highest CVSS and severity level to lowest CVSS and severity level. This is very helpful in determining which vulnerabilities should be prioritized first without having to specify a filter or scroll endlessly! Another interesting finding is there is a column called Exploits. Since Nexpose has the ability to integrate with Metasploit, an open source penetration testing framework also offered by Rapid7, this allows for additional benefits in visibility as well as user convenience since both resources can work in tandem under the same developer. We can see that the first vulnerability with title "ISC Bind: Assertion Failure in buffer.c While Building Response to a Specifically Constructed Request (CVE-2016-2776)" can be exploited with one or more Metasploit modules. There is also a column called "Published On" which is useful to know the date when the vulnerability was officially published and made aware to Rapid7. Knowing how long a vulnerability has been publicly known is a good reference point because it allows for more insight to be extracted. There will be much more information in a vulnerability that has been published for quite some time than a vulnerability that has been only published for a few weeks or few months. In some ways this is both good and bad because although a vulnerability has been published way far back years ago like in this case, this does not mean that the vulnerability no longer poses a threat or should not be of concern anymore! Both old and new vulnerabilities need to be treated the same! The reason for this is because the exploits become publicly known meaning someone ethical or malicious can see how the exploit works. The significant time that a vulnerability has been publicly known is something worth addressing because this means that many users have already been aware of how the vulnerability can be possibly exploited. Even if a patch were made available there still needs to be consideration that every and any vulnerability can damage an organization and its assets and even entire network if taken lightly. For example when clicking on the Metasploit icon in the Exploits column, I am able to identify the existing threats for the vulnerability in the form of exploits. The Source Link column shows where the exploits have been published for the vulnerability. Next to this column is the Exploit Skill Needed column. This is helpful in determining remediation tasks on the most critical gaps of our security. For example it increases both our awareness and visibility in finding out whether for the discovered vulnerability there is associated exploit. The exploit skill needed should be something to keep in mind. As discussed in our paper in the beginning, although the required skill level for the exploit is shown, security cannot be overlooked regardless whether the exploit poses very little harm or whether the exploit skill level is beginner or any skill level for that matter! When we clicked on the Source Link of Exploit Database for the vulnerability we were taken to the Exploit Database website. There is the ability to download the exploit as well as review the source code. The Exploit Database is a fantastic resource for penetration testing and vulnerability assessment. There are even papers written by various security researchers for various platforms which further goes into detail on how the exploits were discovered and later tested. PoC or Proof Of Concept is an exploit which is a non harmful attack against a computer or network. There are examples of PoC in the exploit database which goes to show how beneficial this resource is in

vulnerability assessment. They are explained in great detail which further provides more clarity into how these exploits can be exploited and used for malicious purposes. If one cannot find anything on Google regarding an exploit for a vulnerability then the Exploit Database is your best friend! Our main focus after assessment is to focus on the aspects from both Exploit Database and NIST's National Vulnerability Database that provide benefits and satisfy our criteria in whether Nexpose is sufficient enough in addressing our problem if there is enough visibility in our infrastructure.

VULNERABILITIES & THREATS SECTION FOR ASSET FIOS_QUANTUM_GATEWAY.FIOS-ROUTER.HOME

- Vulnerabilities are organized based on their criticality (CVSS and Risk Score) so that informed decisions can be made in assessment and remediation. Higher CVSS vulnerabilities and risk scores are placed at the top. While those with low CVSS and risk scores are placed at the bottom.
- Actions taken here were to investigate the first exploit "ISC BIND 9 - Denial of Service" from the Exploits column of the vulnerability that is being assessed.
- A source link for both exploits are provided. They are links to Exploit Database and Metasploit Module which is another vulnerability and exploit database resource offered by Rapid7.
- The Exploit Skill Needed column is helpful in determining how feasible the exploit can be exploited. For example exploits with a low skill level needed can be exploited much more easily than an exploit with an intermediate or high skill level needed.

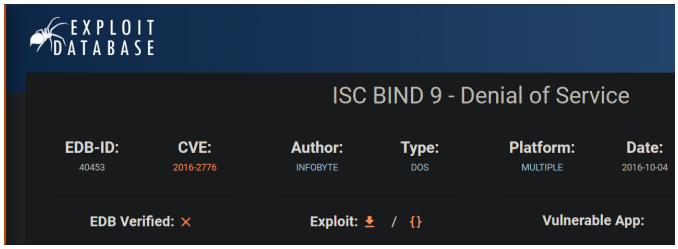




IX . EXPLOIT DATABASE & NIST'S NVD

In the Exploit Database for the "ISC BIND 9 - Denial of Service" exploit we come across several pieces of information. The CVE identifier is given as CVE 2016-2776. The first identifier 2016 lets us know that the vulnerability was published in the year 2016. There is even a hyperlink when you click on the CVE which brings us to NIST's National Vulnerability Database (NVD). This is an excellent resource because it's from a trusted prestigious institute. Not to say that Exploit Database is not trustworthy. We felt that having additional reference especially from an entity like NIST goes to show how important it is to look at all angles when researching an exploit or vulnerability. The NVD provided even more links to advisories published from multiple security and tech companies. This makes us feel very confident because there may be information that was not covered extensively by NIST where as for an advisory from a company like Red Hat or even Microsoft can offer much more extensive valuable information because of the published advisory. An advisory as the name suggests is a public announcement about a security problem either vulnerability or exploit. They offer much more actionable information which can come in very handy when time is of the essence. For example they can provide actual links or downloads to a patch or security update which can help resolve the problem of having a vulnerable computer on a network.

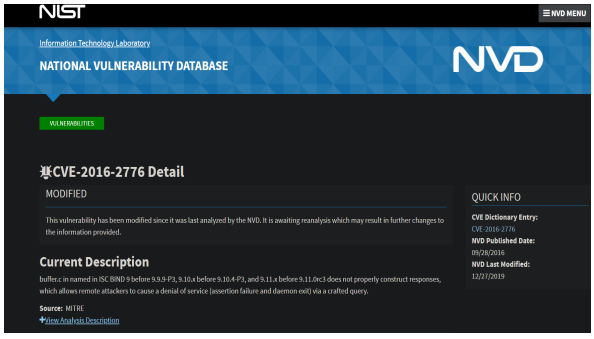
EXPLOIT DATABASE ENTRY



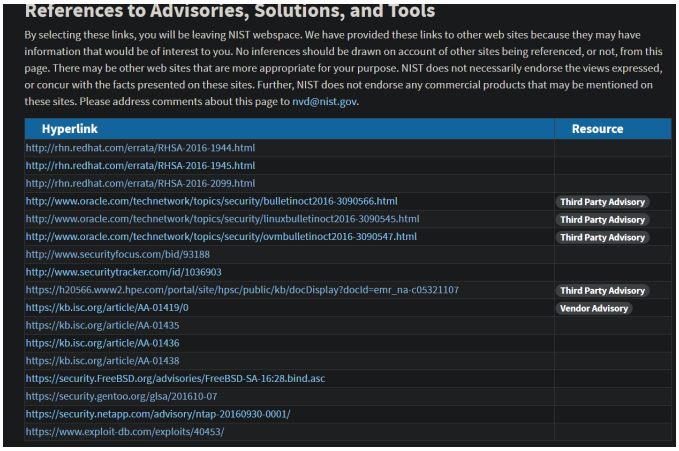
- In addition to the CVE identifier we can also see the EDB-ID which can be used to easily search for the exploit in Exploit Database.
- Information about other published exploits from the author can be found.
- The type of exploit is listed as DOS. When clicked we can see all the DOS type of exploits in Exploit Database.
- Lastly there is the ability to download and view raw the exploit. The exploit was created using Python. Below are just some lines of code from the exploit.

```
import socket
import struct

TARGET = ('192.168.200.10', 53)

Q_A = 1
Q_TSIG = 250
DNS_MESSAGE_HEADERLEN = 12


def build_bind_nuke(question="\x06google\x03com\x00", udpsize=512):
    query_A = "\x8f\x65\x00\x00\x00\x01\x00\x00\x00\x00\x00\x01" + question + int16(Q_A) + "\x00\x0

    sweet_spot = udpsize - DNS_MESSAGE_HEADERLEN + 1
    tsig_rr = build_tsig_rr(sweet_spot)

    return query_A + tsig_rr

def int16(n):
    return struct.pack("!H", n)

def build_tsig_rr(bind_demarshalled_size):
    signature_data = ("\x00\x00\x57\xeb\x80\x14\x01\x2c\x00\x10\xd2\x2b\x32\x13\xb0\x09"
                      "\x46\x34\x21\x39\x58\x62\xf3\xd5\x9c\x8b\x8f\x65\x00\x00\x00\x00")
    tsig_rr_extra_fields = "\x00\xff\x00\x00\x00\x00"

    necessary_bytes  = len(signature_data) + len(tsig_rr_extra_fields)
    necessary_bytes += 2 + 2 # length fields

    # from sizeof(TSIG RR) bytes conforming the TSIG RR
    # bind9 uses sizeof(TSIG RR) - 16 to build its own
    sign_name, algo_name = generate_padding(bind_demarshalled_size - necessary_bytes + 16)

    tsig_hdr = sign_name + int16(Q_TSIG) + tsig_rr_extra_fields
    tsig_data = algo_name + signature_data
    return tsig_hdr + int16(len(tsig_data)) + tsig_data

def generate_padding(n):
    max_per_bucket = [0x3f, 0x3f, 0x3f, 0x3d, 0x3f, 0x3f, 0x3f, 0x3d]
    buckets = [1] * len(max_per_bucket)

    min_size = len(buckets) * 2 + 2 # 2 bytes for every bucket plus each null byte
    max_size = sum(max_per_bucket) + len(buckets) + 2

    if not(min_size <= n <= max_size):
        raise RuntimeException("unsupported amount of bytes")
```

NIST'S NATIONAL VULNERABILITY DATABASE



ADDITIONAL RESOURCES IN THE NVD (ADVISORIES, SOLUTIONS, & TOOLS) FOR CVE-2016-2776



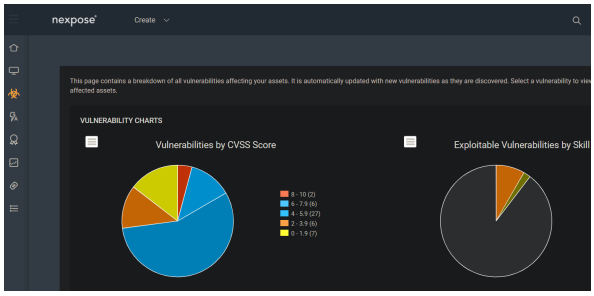
References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink	Resource
http://rhn.redhat.com/errata/RHSA-2016-1944.html	
http://rhn.redhat.com/errata/RHSA-2016-1945.html	
http://rhn.redhat.com/errata/RHSA-2016-2099.html	
http://www.oracle.com/technetwork/topics/security/bulletinoct2016-3090566.html	Third Party Advisory
http://www.oracle.com/technetwork/topics/security/linuxbulletinoct2016-3090545.html	Third Party Advisory
http://www.oracle.com/technetwork/topics/security/ovmbulletinoct2016-3090547.html	Third Party Advisory
http://www.securityfocus.com/bid/93188	
http://www.securitytracker.com/id/1036903	
https://h20566.www2.hpe.com/portal/site/hpsc/public/kb/docDisplay?docId=emr_na-c05321107	Third Party Advisory
https://kb.isc.org/article/AA-01419/0	Vendor Advisory
https://kb.isc.org/article/AA-01435	
https://kb.isc.org/article/AA-01436	
https://kb.isc.org/article/AA-01438	
https://security.FreeBSD.org/advisories/FreeBSD-SA-16:28.bind.asc	
https://security.gentoo.org/glsa/201610-07	
https://security.netapp.com/advisory/ntap-20160930-0001/	
https://www.exploit-db.com/exploits/40453/	

With this information we strongly believe that using Exploit Database as a reference together with Nexpose and NIST's NVD provides enhanced visibility. Some key findings were that this vulnerability cannot be exploited if the ISC BIND is upgraded to the latest version. Nexpose offers remediations in the form of vulnerability rollup solutions and vulnerability solutions. The difference between the two solutions is that the rollup solutions are of the highest supersedence. This means that these solutions are the most recent and broad which supersedes those that are just regular solutions. Vulnerability solutions offers the full list of available solutions for the vulnerability with no determination as to which has the highest supersedence. A rollup solution can be essential when time is of the essence especially when a vulnerability has been ranked with a critical and high CVSS.score. This allows for efficient remediation in terms of speed and confidence that the provided solution will be of much greater benefit in remediating the vulnerability as a whole without having to rely on additional solutions! However there may be a specific vulnerability that may require more than 1 solution. In this scenario a System Administrator needs to decide whether the vulnerability should be remediated right away or not. Taking a look at the vulnerabilities page vulnerability charts we can see two charts.

VULNERABILITY CHARTS



- Vulnerabilities are displayed by their CVSS score and exploitable skill levels.
- The CVSS score chart displays how many vulnerabilities fall into each of the CVSS score ranges. The score is based on access complexity, required authentication, and impact on data. Score ranges from 1-10, 10 being the worst so prioritizing the vulnerabilities with higher numbers should come first.
- Over half (56.25%) of the vulnerabilities discovered had a CVSS score range of 4-5.9 (27 vulnerabilities scored this range). This was the highest percentage in our chart.
- However the most concerning part in this chart is the percentage of critical vulnerabilities that were discovered. A percentage of 4.17% which comprised 2 vulnerabilities scored a CVSS range between 8-10.

Another factor to consider here is the Exploitable Vulnerabilities by Skill Level chart. This chart categorizes the vulnerabilities by their level of skill required to exploit them. A key determination we made is that those vulnerabilities that are categorized as easily exploitable present the greatest threat. The reason is because there will be more people who possess the necessary skills. This sets a good guideline for the entire vulnerability remediation phase. Remediating vulnerabilities that are considered easily exploitable in terms of skill level should be remediated first. Once these vulnerabilities have been remediated then working your way up from skill level to skill level is a good approach here. Having a consideration in both exploit skill level and CVSS score is fundamental in our approach in handling vulnerabilities as well as addressing visibility in our infrastructure.

These charts provide justification on which vulnerabilities to prioritize to secure the infrastructure as efficiently and quickly as possible. This effectively satisfies our problem statement.

X . CONCLUSION

Increasing visibility in an IT infrastructure is a multi faceted process. There needs to be significant work involved from scanning to assessment and prioritization of vulnerabilities. We have learned that just using Nexpose is not sufficient in terms of remediating vulnerabilities. Even though Nexpose offers additional resources like links to NIST's NVD and Exploit Database it is not sufficient. There needs to be more input from other sources like security researchers and professionals. We mentioned how Exploit Database offers actual papers addressing specific exploits in great detail and some even presenting a PoC of the exploit freely available to download. We also mentioned how companies like Red Hat, or Microsoft and other top companies should continue to offer guidance to those entities that lack the resources. This guidance can be in the form of advisories or remediation steps like an actual security update or patch. Nexpose taught us that vulnerability assessment and intrusion detection cannot be handled simply by using fancy tools. There needs to be a team effort in each phase from assessment to remediation. Nexpose made us feel very confident that although we were just beginning to learn the tool, we were given sufficient resources and input in making difficult decisions like for example which vulnerabilities to prioritize and why. The vulnerability charts gave us a graphical representation of what our infrastructure faced in terms of threats. It's almost as if the threat landscape was presented to us and with this information we were given the list of vulnerabilities based on their criticality in terms of its CVSS and risk score. This was a feature I enjoyed very much. Although we were unable to use the aggressive discovery scan to see whether our scan would have produced different results, I am very

confident that the visibility in our infrastructure has been made much more clearer in terms of what devices have vulnerabilities and which ones present the greatest threat to our infrastructure.

REFERENCES

[1] 4min. read, “What is Zero Trust?,” *What is Zero Trust? - Palo Alto Networks*. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-a-zero-trust-architecture. [Accessed: 4-Jun-2020].

[2] Rapid7 Blog. 2020. *Insightidr & Nexpose Integrate For Total User & Asset Security Visibility*. [online] Available at: <https://blog.rapid7.com/2016/09/22/insightidr-nexpose-integrate-for-total-user-asset-security-visibility/> [Accessed 4 June 2020].

[3] Nexpose. 2020. *Using Exploit Exposure*. [online] Available at: <https://nexpose.help.rapid7.com/docs/using-exploit-exposure> [Accessed 4 June 2020].

[4] Cve.mitre.org. 2020. *CVE -Common Vulnerabilities And Exposures (CVE)*. [online] Available at: <https://cve.mitre.org/> [Accessed 4 June 2020].

[5] SearchSecurity. 2020. *What Is CVSS (Common Vulnerability Scoring System)? - Definition From Whatis.Com*. [online] Available at: <https://searchsecurity.techtarget.com/definition/CVSS-Common-Vulnerability-Scoring-System> [Accessed 25 June 2020].

[6] Exploit Database. 2020. *ISC BIND 9 - Denial Of Service*. [online] Available at: <https://www.exploit-db.com/exploits/40453> [Accessed 25 June 2020].

[7] Nvd.nist.gov. 2020. *NVD - CVE-2016-2776*. [online] Available at: <https://nvd.nist.gov/vuln/detail/CVE-2016-2776> [Accessed 25 June 2020].

[8] Exploit-db.com. 2020. *Offensive Security’S Exploit Database Archive*. [online] Available at: <https://www.exploit-db.com/papers> [Accessed 25 June 2020].

[9] Rapid7 Blog. 2020. *Nexpose Community Edition Lab | Scanning & Reports*. [online] Available at: <https://blog.rapid7.com/2012/09/19/using-nexpose-at-home-scanning-reports/> [Accessed 25 June 2020].

[10] Nexpose. 2020. *Working With Vulnerabilities*. [online] Available at: <https://nexpose.help.rapid7.com/docs/working-with-vulnerabilities> [Accessed 25 June 2020].