strobes™

White Paper

Prioritization Score for Vulnerabilities

This white paper introduces the Strobe Bug prioritization score, a data-driven approach to vulnerability management. Learn how to objectively prioritize risks based on severity and potential impact, allowing you to focus on the vulnerabilities that matter most.

Contents

1. Introduction

One of the greatest challenges in the rapidly growing IT industry right now is to make sure that all the code written for the services is secure. One of the ways to make sure that the code of a service is secure by trying to exploit the service.
For which we can use 2 methods:

- First is manual testing of the service.
- Second is using automated scans which checks for vulnerabilities.

Whether you choose any of the 2 methods or their combination one thing that is a definite is that there will be some vulnerabilities in the product.

The second challenge now is to find which vulnerability to patch first because we can find a lot of vulnerabilities in the scan and patching all the vulnerabilities is a very resource intensive process. This is where prioritization of vulnerabilities comes into picture because developers only need to fix the vulnerabilities that have a high chance of getting exploit. Prioritization score is a method of scoring vulnerabilities based on multiple factors to give out a score for each vulnerability which indicates the urgency with which that vulnerability should be fixed.

2. What is the need for a prioritization score.

According to cvedetails.com there were 18325 vulnerabilities were released in 2020 that is around 352 vulnerabilities per week and this number is for the vulnerabilities that have been indexed and documented by MITRE but there are other zero day vulnerabilities that exist in the wild and are being exploited. So, to do the examination and patching of vulnerability and then testing it for all the vulnerability is not an easy task and that is why we need an effective prioritization system which can prioritize the vulnerabilities which pose actual danger to the organizations and weed out the vulnerabilities that either have a very low chance of getting exploited or cannot do much damage to the integrity of the system if exploited.

3. Mathematical Approach to Vulnerability management.

To find all the vulnerability that are going to be exploited we need a mathematical approach to make a equation that can predict based on different factors of the vulnerability. Today industry standard practice is to use Common Vulnerability Scoring System (CVSS) rating of 1 (lowest) to 10 (most severe) but in 2013 an article published by Mike Roytman of Kenna Security and Dan Geer of In-Q-Tel showed that "a high CVSS score does not imply impending risk in need of immediate mitigation.". That means we need a system which takes factors into account which are better indicator of impending attack or exploitation.

4. Strobes bug prioritization score for vulnerabilities.

While looking at the vulnerabilities that are in the wild, we can observe that a high level of vulnerability on a asset/resource is not worth patch first if that asset does not contain sensitive material or is connected to another such asset. Similarly, a low-level vulnerability should be prioritized if it is on an asset/system that contains highly sensitive material like user information. To take this factor into account we divide the prioritization based on two components:

- Vulnerability
- Asset

A. Vulnerability

According to Wikipedia “a vulnerability is a weakness which can be exploited by a threat actor, such as an attacker, to cross privilege boundaries (i.e. perform unauthorized actions) within a computer system”. We can characterize the level of the vulnerability by looking at what is needed to exploit this vulnerability? For eg:

- Do we need physical access to the system?
- Is there any tool already written to exploit this vulnerability?
- Do we need to write code from scratch to exploit it?
- Is there any patch released by the vendor for the same vulnerability? Etc

and What actions can we do after successfully exploiting the vulnerability? For eg:

- Can we execute code or instructions on the exploited asset?
- Can we gain administrative privileges?
- Can we make the asset unavailable for the user (Dos)? Etc

Now let’s start to convert these factors into weightages. The strobes bug prioritization score ranges from 0-1000 (0 being the lowest and 1000 being the most severe). Now let’s first look at the vulnerability part.
Because most vulnerabilities are indexed and most of their characteristics are already available online by MITRE we will use those features.

A1. CVSS Score

We take CVSS as a feature for our score as well because there are many factors that are considered in the CVSS that are relevant. So we give out around 600 weightage out of 1000 to the CVSS that means that if a vulnerability has 10 CVSS it will score 600 /600 in our score system but it is not a linear function because even thought the 10 cvss will score 600 and 0 cvss will score 0 but if I take cvss 8 it should be around 557and cvss 5 should be around 42 because as the cvss gets below 8 it is observed that the vulnerability is very rarely a sever one. So, we have fitted this data into an equation and use that to get the value of the score based on the cvss.

CVSS	Score out of 555
0	0
5	42
6	179

8	557
10	600

A2. CWE Score

After CVSS score the next factor that comes is the CWE of the vulnerability that tell us what will be the impact of the vulnerability if it is successfully exploited. For eg: whether on exploitation the vulnerability will give the attacker the ability to temporarily down the service or whether the attacker will gain permission to execute any code that they want. So based on the different impact we will score the vulnerability different score.

The total weightage pf a vulnerability can get is around 220 that means a vulnerability with highest impact will get 220 score. Now some vulnerabilities have more than one CWE some do not have a CWE attached to it so for the vulnerabilities which have more than one CWE attached to it we will consider the vulnerability with higher impact and vulnerabilities which do not have a CWE attached to them will have a default value.

To decide which impact is higher we will assign different magnitude for eg:

Impact	Magnitude
Execute Unauthorized Code or Commands	0.95
Bypass Protection Mechanism	0.8
DoS	0.7
Reduce Maintainability	0.4
Other	0.2

To get the score we just multiply the magnitude by 220.

A3. Seen Wild

When a vulnerability is being exploited by the masses it can be said that the vulnerability is being seen in the wild it generally happens when the exploitation of the vulnerability does not require much skills (either it is easy to exploit or a tool is made to exploit it.) or it is harder to patch the vulnerability.

Weightage of the seen wild is 40 points in the score.

The weightage of the vulnerability for seen wild:

Seen Wild	Magnitude
True	0.91
False	0.23

To get the score we just multiply the magnitude to 40.

A4. Privilege Required

To exploit a vulnerability the attacker should have a level of privilege to exploit the vulnerability so the attacker can either use another vulnerability that gives the attacker escalated privileges or can use a vulnerability that does not require high privileges to exploit for example if a vulnerability that can execute code on the victims machine but all the important data that the attacker wants is secured by admin/root account then the attacker will need higher permission to get that data.

We can use MITRE CVE information to find what level of privilege does the vulnerability need to be successfully exploited.

Weightage of the privilege required is 20 points in the score.

The weightage of the vulnerability for seen wild:

Seen Wild	Magnitude
None	0.85
Low	0.62
High	0.27

To get the score we just multiply the magnitude to 20.

A5. Exploit Available

When a vulnerability has a ready to use tool or a script which is released in the open and can be used by people with moderate skills that means that the vulnerability has an exploit available. This is important because if a tool is available to exploit and the software version being used is not patched for that vulnerability then the number of potential attackers exponentially increase with the release of the script/tool.

To find if the tool or script is released, we can check multiple feeds that keep a list of exploits like exploitDB, Metasploit etc.

Weightage of the exploit available is 20 points in the score.

The weightage of the vulnerability for exploit available:

Seen Wild	Magnitude
True	0.71
False	0.25

To get the score we just multiply the magnitude to 20.

A6. User Interaction

User interaction as name suggest means how much interaction to the victim’s asset is needed to exploit the vulnerability or in other words does the attacker need the user to perform some action for the attacker to be

able to exploit the victim's asset. The action can be anything from the user clicking a link to plugging and usb to the machine.

We can use MITRE CVE information to find what level of interaction does the vulnerability need to be successfully exploited.

Weightage of the user interaction is 20 points in the score.

The weightage of the vulnerability for user interaction:

User Interaction	Magnitude
None	0.85
Required	0.62

To get the score we just multiply the magnitude to 20.

A7. Malware Available

A malware is a piece of code that is created to perform actions with a malicious intent. A malware is mostly a modified or engineered exploit that exploits a vulnerability. A simple example will be ransomware using a code execution vulnerability.

We can use data from dark web forums and marketplaces to find whether or not a vulnerability has a malware available or not.

Weightage of the malware available is 15 points in the score.

The weightage of the vulnerability for malware available:

Malware Available	Magnitude
True	0.42
False	0.27

To get the score we just multiply the magnitude to 15.

A8. Patch Released

Whenever a vulnerability is found the vendor, whose product is affected by the vulnerability starts working on the patch for the product so that it becomes safe from the exploitation from that vulnerability so the patch released factor comes as negative in our score.

We can use references of NVD page from CVE to see if the vendor released any patch for the vulnerability.

Weightage of the patch released is negative 10 points in the score.

The score for the patch is low because in most commercial setting updating the version of the affected software is not immediate so it is assumed that most of the assets remain vulnerable even after the patch is released

The weightage of the vulnerability for patch released:

Patch Released	Magnitude
True	0.71
False	0.15

To get the score we just multiply the magnitude to 10.

A9. Vulnerability Exploitation Probability Score

All the parts of the prioritization score till now are static which means they do not change except seen wild, exploit and malware available features. So, to add the dynamic part to the score we added vulnerability exploitation probability which uses Exploit Prediction Scoring System (EPSS).

There are multiple factors that affect this score:

1. Vendor: The vulnerability affected products belong to which vendor. This is an important factor as all the companies react to the vulnerabilities differently, for some companies a very small vulnerability should be immediately fix because they are a huge organization and do not know which vulnerability can make a chain vulnerabilities possible. On the other hand, there are smaller organizations that do not have the resources to tackle each vulnerability so they only focus on the critical ones. This model gives weight to the vulnerability based on the vendor it affects.

    a. Microsoft 2.44
    b. IBM 2.07
    c. Adobe 1.91
    d. HP 1.62
    e. Apache 1.1
    f. Google -0.89
    g. Apple -1.92
    h. Other web services 0.06

2. Vulnerability Type: The vulnerability type tells us what kind of actions the attacker can perform if they can successfully exploit the vulnerability for example remote code execution can help attacker find info about customers from an e-commerce website whereas a dos vulnerability can stop the user from visiting the website. Based on the severity of the actions this algorithm gives weight to the vulnerability.

    a. Arbitrary code execution 0.57
    b. Remote code execution 0.23
    c. Denial of Service 0.22
    d. Memory Corruption -0.2

3. Exploit Available: The severity of the vulnerability increases with the decrease in the complexity to use the tool used to exploit it. It is harder to exploit a vulnerability if you have to write code from scratch to exploit this vulnerability and much easier if you find a script to do it online, and even easier if you find a tool with graphical user interface which only requires you to input certain parameters about the target. You can use these tools to exploit even if you do not know how the vulnerability is actually being exploited. Based on the tool's availability algorithm gives weight to the vulnerability.

    a. Exploit available 2
    b. Proof of concept 1.5

4. References: the vendor reference's part makes it dynamic because as the CVE gets old the vendor references increase which includes patch report, new version release etc. Generally, the higher the number of the references means the more attention the CVE has gotten from the vendor meaning they are releasing cautionary documents as to avoid certain versions of the product or releasing a patch for the vulnerability.

    a. Log(no_of_ref +1) * 1.01

5. Level of access: If the attacker needs physical access to the machine to exploit it becomes harder for the attacker to exploit the vulnerability due to the additional security that they will have to breach in to reach the machine therefor a weight is also attached based on the level of access needed to exploit the vulnerability.

    a. Can only be exploit locally -0.63

To get the probability value we add the all the weights applicable for the CVE to the base of -6.18 and then use that weight number n in the formula $1 / (1 + e^{-n})$.

To get the score from that probability we multiply the probability to 100.

Example:

CVE-2017-0143

CVE- Component	Value	Weightage	Maximum Score	Score
CVSS	8.1	N/A	600	563.7
CWE	Code Execution	0.95	220	211.2
Seen_Wild	True	0.91	40	36.4
Priv_Required	None	0.85	20	17.0
Exploit Available	True	0.71	20	14.2
User Interaction	None	0.85	20	17.0
Malware Available	True	0.42	15	6.3
Patch Released	True	0.71	-10	-7.01
Exploitation Probability	0.931	N/A	100	93.1

Here If we add all the scores, we get 951.8/1025 but if we see that by multiply weightage to max score we are losing some points even if the vulnerability scores should be maximum but it is limited due to the max weightage. According the weightage a vulnerability can max score 999 so this score is actually 951.8/999.

B. Asset

Here an asset can be any system that is running any kind of software it can be a router, a server instance, docker container etc. To categorize the sensitivity level of the asset we can look at certain properties of the asset such as:

- The sensitivity of the data stored on that asset
- Is there any high-level asset connected to the asset that has the vulnerability?
- Does the asset accept connection from pubic IP (i.e. public facing)?

B1. Asset Sensitivity

Asset Sensitivity can be defined as the importance of the asset which is decide by the data that is present on the asset or the sensitivity of the other assets that are connected to this asset. So if an asset has confidential

data or if it is connected to an other asset which contains sensitive data than we can say the sensitivity of the asset is high.

The weightage of the vulnerability for asset sensitivity:

Asset Sensitivity	Magnitude
None	0
Low	1.23
Medium	2.523
High	3.2
Critical	4

To get the score we just multiply the magnitude to 0.225.

B2. Asset Exposure

Asset exposure tells us that whether or not the asset can be accessed from anywhere or it can only be accessed through private network. So, if the service on the asset is available for everyone then it is public otherwise private. The asset which is public facing is more prone to attacks.

The weightage of the vulnerability for Asset exposure:

Asset exposure	Magnitude
Public	0
Private	-0.07

B3. SLA Passed

SLA is the timeline to patch the asset so SLA is passed if the date to patch the asset has been passed but the patch is still not released.

Users have to provide this data about the asset to calculate more precise score.

The weightage of the vulnerability for SLA Passed:

SLA Passed	Magnitude
True	0.02
False	0

B4. Intercept

Intercept is a value to adjust total value of the equation. We add this offset to the sum of weightage (Asset sensitivity, Asset exposure, Sla Passed).

Value of the Intercept is 0.08

Example (Cont.):

Let's continue the example CVE-2017-0143 where the vulnerability scores 951.8/999. Now let's assume the user gave us the information that the asset on which this vulnerability has been found is Public, has Medium Sensitivity and Its patch is still to be deployed but the SLA has not been passed. So, the asset score will be.

Asset- Component	Value	Numeric Value	Weightage	Score
Asset Sensitivity	Medium	2	0.225	0.445
Asset Exposure	Public	0	-0.07	0
SLA Passed	False	0	0.02	0
Intercept	N/A	1	0.08	0.08

The asset weightage comes out to be 0.445 + 0 + 0 + 0.08 comes out to be 0.53. So, the final prioritization score will be CVE score * Asset Weightage = 951.8 * 0.53 => 504/999

C. Summary

Today the cyber-attacks are increasing more and more in every industry and the need of the hour is for companies to prioritize on vulnerability management and patching. Not all companies have the resources to patch all the vulnerabilities that they find in times when every day we are finding new vulnerabilities. Products like strobes will be able to help industries in fulfilling their cyber security needs. Right now, automated prioritizing of vulnerabilities is the only scalable method that can be used to fight against these cyber threats.

5. Asset Risk Score

The prioritization score is useful for calculating the priority on the bug level but when we look at an organization there are a lot of bugs present in all the different assets. So, we need something for the macro level so that we can see the state of the organization at a glance. For exactly this purpose we use asset risk score. The asset risk scores take all the vulnerabilities that are present in the asset into consideration and mix it with the recently sla violated and patched vulnerabilities and gives out a number out of 100 that represents the risk that a particular asset present to the organization. So, the higher the number the higher should be its priority in vulnerability patching.

Asset score takes 2 scores into consideration.

5A. Asset Risk.

- The asset risk suggests the risk that the asset has if its vulnerabilities are not patched. To calculate this score, we divide the vulnerabilities into 4 severities based on prioritization scores.
  1. >900: Critical
  2. 750-900: High
  3. 550-750: Medium
  4. <550: Low

Then we calculate the number of vulnerabilities the asset has for these 4 categories.

- Second metric that we need is the Asset Sensitivity:
  The sensitivity of the asset can be "Critical"," High", "Medium", "Low" based on the asset sensitivity the score is calculated.

- Third metric is the number of vulnerabilities for which the sla is violated.
  1. Number of critical vulnerabilities with sla violated.
  2. Number of high vulnerabilities with sla violated.
  3. Number of medium vulnerabilities with sla violated.

4. Number of low vulnerabilities with sla violated.

- The last metric that we need is the asset exposure which can be either “Private” or “Public”

After getting these 4 metrics we use different non linear regression equations to find the score for each metric separately.
For example, the metric values for an asset are:

Number of Critical Vulnerabilities:	1
Number of Critical Vulnerabilities:	2
Number of Critical Vulnerabilities:	51
Number of Critical Vulnerabilities:	243
Asset Sensitivity:	"Critical"
Number of Critical Vulnerabilities with sla violated:	0
Number of Critical Vulnerabilities with sla violated:	0
Number of Critical Vulnerabilities with sla violated:	10
Number of Critical Vulnerabilities with sla violated:	24
Asset Exposure:	“Public”

Using the equation, the risk score comes out to be **98**

5B. Patch Efforts Score

The patch efforts score represents the factor or pace with which the vulnerabilities are getting patched. Based on the efforts that are being put towards patching the Asset risk score can come down by 0-6%. This score metric uses the number of vulnerabilities patched of each type in the past 7 days to calculate a factor and that factor determines the reduction in total score.

Based on the number of vulnerabilities patched the score can be reduced to a point and obviously patching of high severity vulnerability results in more reduction in risk score.

To continue the above example with the metrics:

Number of critical vulnerabilities patched in last 7 days:	4
Number of high vulnerabilities patched in last 7 days:	3
Number of medium vulnerabilities patched in last 7 days:	12
Number of low vulnerabilities patched in last 7 days:	14

Putting this value in equation gives out the result 0.964

So, the total score is, risk *score – (risk_score * 0.05 * patch_effort_factor)* => 98-(98*0.05*0.964) => **93.3**