Optimizing Microsoft security investment with vulnerability data to prevent botnet infections

Xander Bouwman

Paul van der Knaap

Daan van der Valk

Dereck Bridie

Gouri Calamur Viravalli

WM0824TU Group 1

Introduction

Microsoft is one of the largest IT companies in the world, whose products and services are diverse and widespread. Microsoft Windows and Office are the most used desktop operating system and office suite, respectively. In addition, Microsoft's internet browsers Internet Explorer and Edge, as well as their operating systems and their office solutions all have a significant market share. As billions of people worldwide make use of Microsoft products, expecting a safe and secure experience, the investments on making and keeping systems secure is a major concern.

The security issue addressed in this paper is botnet infections. Botnets are one of today's key challenges in cyber security. Botnets are collections of infected devices connected to the Internet. Once a device is infected with botnet software, it can be used to perform distributed denial-of-service (DDoS) attacks, send spam, and steal data. Often, this happens in background processes, without the knowledge of the device owner.

Botnets are collections of maliciously infected computers (or bots) that are used by cyber criminals as tools for criminal activities. When a cybercriminal has created a large network of bots, it can be used for DDoS attacks on computers or servers connected to the Internet. These DDoS attacks can be harmful in many ways, causing downtime that leads to loss of profit and other damages.

The attackers can use vulnerabilities in software to infect computers connected to networks and to force them to join this botnet. However, these vulnerabilities also often cause damage to the device’s owner as well. An infected device’s data is also exposed. The device owner’s assets are also at risk after such an infection: their private data can be mined and sold.

Microsoft has a couple of reasons to proactively try to prevent botnet infections:

- As one of the top IT vendors, Microsoft has great responsibility in producing cyber secure systems as they can affect both consumers and businesses worldwide.
- Although it is hard to define exact security levels, nowadays both companies and individuals consider security as important issues when deciding which products they buy.
- As a company that provides cloud services, Microsoft would like to avoid getting DDoSed by their own systems.
- With modern-day malware used for botnets, users often don't even notice their PC is infected. Users therefore have no incentive to remove the malware, even though the collective effect of malware-infected systems DDoS'ing is harmful and costly for the victims. This means the IT sector has to take responsibility to battle the issue of botnets.

Despite a motivation to act against botnet infections, however, it is a challenge to put resources to their best use when battling them.

In business, IT Management in general is responsible for IT vision and strategy, allocation of technical resources and internal IT policy. General IT Management, led by the Chief Technology Officer, decides which IT infrastructures, services and products to buy externally, and which IT artifacts to develop internally. For cost-benefit analyses of external IT products and services, vulnerability data could help decide which products and services to buy. For internal IT projects, vulnerability data could indicate points of attention during development to avoid certain flaws.

Cyber Security Management in particular, led by the Chief (Information) Security Officer, is responsible for the enterprise's assets protection, and the corresponding vision and strategy. The key performance indicators are all about vulnerabilities and other aspects of risk (like treat and impact). Metrics should assist in allocating resources for security by establishing the security strategy, assessing the security performance, and prioritizing issues.

Research Question

The research question we want to answer in this paper is the following:

Suppose we are security policy makers working for Microsoft. How should additional security budget be distributed to prevent botnet infections under our users?

Method

We have been assigned the National Vulnerability Database dataset[1]. The National Vulnerability Database (NVD) is the repository used by the National Institute of Standards and Technology (NIST), which is the US government agency responsible for cybersecurity. This project makes use of the NVD's Common Vulnerabilities and Exposure (CVE) lists for the years 2010 until September 17[2].

The data reports vulnerabilities in software that could potentially lead to botnet infections, with a corresponding severity that indicates the required action to be taken by system administrators. Although vulnerabilities provide varying levels of access to the host machine, we will assume that every vulnerability could be used to infect the host into a botnet

This dataset is a list of CVEs. A CVE defines a vulnerability as such:[3]

"A weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability. Mitigation of the vulnerabilities in this context typically involves coding changes, but could also include specification changes or even specification deprecations (e.g., removal of affected protocols or functionality in their entirety)."

A single given CVE has the following information:

- **Affected product information**
  Lists vendor data and product data such as product name and affected versions.
- **References**
  Lists reference data. This could be security bulletin notifications or other sources of information about the CVE.
- **Description**
  Long text description of the CVE. An example is the following:
  Microsoft Internet Explorer 9 through 11 allows remote attackers to bypass the Same Origin Policy via unspecified vectors, aka "Internet Explorer Elevation of Privilege Vulnerability."

[1] https://nvd.nist.gov/vuln/data-feeds
[2] Date range of dataset 1/1/2010 until 25/9/2017
[3] https://nvd.nist.gov/vuln

- **Impact**

  This impact is measured in both CVSSV3[4] and CVSSV2[5]. Because CVSSV3 is an extension of CVSSV2 and contains strictly more data than CVSSV2, we will only use CVSSV3 data in this report. This impact data contains the following, from the guide on CVSSV3[6]:

  - **Attack Vector:**
    This metric reflects the context by which vulnerability exploitation is possible. This metric value (and consequently the Base score) will be larger the more remote (logically, and physically) an attacker can be in order to exploit the vulnerable component. The assumption is that the number of potential attackers for a vulnerability that could be exploited from across the Internet is larger than the number of potential attackers that could exploit a vulnerability requiring physical access to a device.
  - **Attack Complexity:**
    This metric describes the conditions beyond the attacker's control that must exist in order to exploit the vulnerability. As described below, such conditions may require the collection of more information about the target, the presence of certain system configuration settings, or computational exceptions. Importantly, the assessment of this metric excludes any requirements for user interaction in order to exploit the vulnerability (such conditions are captured in the User Interaction metric). This metric value is largest for the least complex attacks.
  - **Privileges Required:**
    This metric describes the level of privileges an attacker must possess before successfully exploiting the vulnerability. This metric is greatest if no privileges are required.
  - **User Interaction:**
    This metric captures the requirement for a user, other than the attacker, to participate in the successful compromise of the vulnerable component. This metric determines whether the vulnerability can be exploited solely at the will of the attacker, or whether a separate user (or user-initiated process) must participate in some manner.
  - **Scope:**
    Formally, Scope refers to the collection of privileges defined by a computing authority (e.g. an application, an operating system, or a sandbox environment) when granting access to computing resources (e.g. files, CPU, memory, etc). These privileges are assigned based on some method of identification and authorization. In some cases, the authorization may be simple or loosely controlled based upon predefined rules or standards.
  - **C/I/A Impact:**
    The Impact metrics refer to the properties of the impacted component. Whether a successfully exploited vulnerability affects one or more components, the impact metrics are scored according to the component that suffers the worst outcome that is most directly and predictably associated with a successful attack. That is, analysts should constrain impacts to a reasonable, final outcome which they are confident an

[4] https://www.first.org/cvss/specification-document
[5] https://www.first.org/cvss/v2/guide
[6] https://www.first.org/cvss/specification-document

attacker is able to achieve.

- **Base Score:**
  The Base Score is a function of the Impact and Exploitability sub score equations.
- **Exploitability Score:**
  The exploitability score is defined as follows:
  8.22 × AttackVector × AttackComplexity × PrivilegeRequired × UserInteraction
- **Impact Score:**
  Scope Unchanged 6.42 × ISC_{Base}
  Scope Changed 7.52 × $[ISC_{Base} - 0.029] - 3.25 \times [ISCBase - 0.02]^{15}$
  Where ISC_{Base} = 1 - [(1 − ImpactConf) × (1 − ImpactInteg) × (1 − ImpactAvail)]

Using the vendor and product information for CVEs, we can filter out products that are not developed by Microsoft. From this smaller dataset, we have chosen a set of products Microsoft distributes to customers. These products are Windows, Internet Explorer/Edge, and Office. The products have been chosen because they are well represented.

Towards metrics

To answer the research question, we will use the NIST NVD vulnerability data to create metrics. These metrics will be vital for policy makers so that decisions can be based on facts. To decide what metrics will be useful to answer this question, we will first look at ideal metrics we can come up with and metrics which are used in practice.

Vulnerability metrics

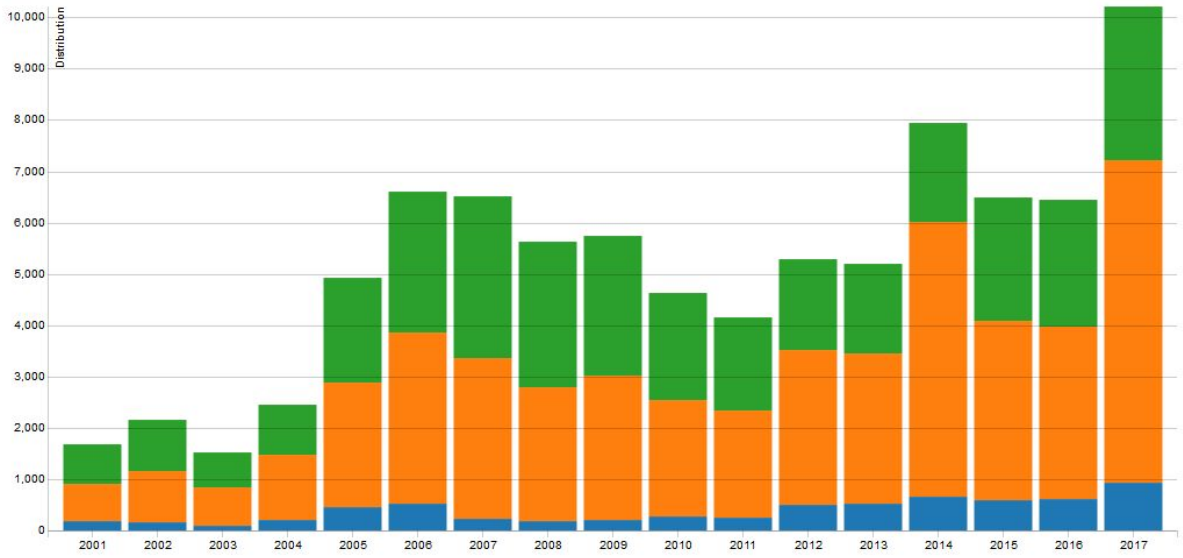
Metrics that we create ideally should provide information on the following issues:

- Which controls should be invested in?
- Where should resources be allocated?
- How do departments perform security-wise? Do they perform better or worse than before?
- What vulnerabilities will have the most impact in the future?
- What software or hardware bugs to particularly look out for during development of IT artifacts?
- Can users be protected from infections using other methods, for example, with a campaign on safety awareness?

Metrics in Practice

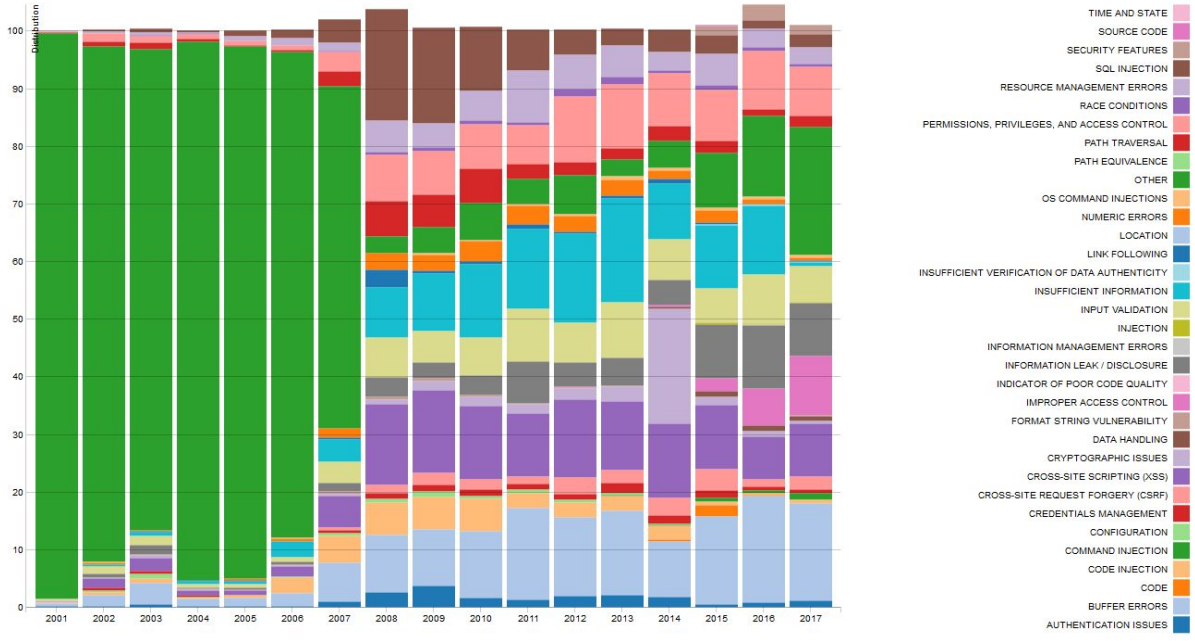
- The Common Vulnerability Scoring System (CVSS) metric combines data in the NVD to give an indication of the severity of a reported vulnerability.[7]

[7] https://nvd.nist.gov/vuln-metrics/cvss



CVSS Severity Distribution Over Time

Source: NIST



Relative Vulnerability Type Totals By Year

Source: NIST

- The Center for Internet Security (CIS) has also established metrics for organizations to use. CIS has divided their metrics into six critical business functions - Incident Management, Vulnerability Management, Patch Management, Configuration Management, Change Management and Application Security.
- The Cybersecurity Framework developed by NIST in 2014 presented five functions as a part of the framework's core - identification, protection, detection, response, and recovery.

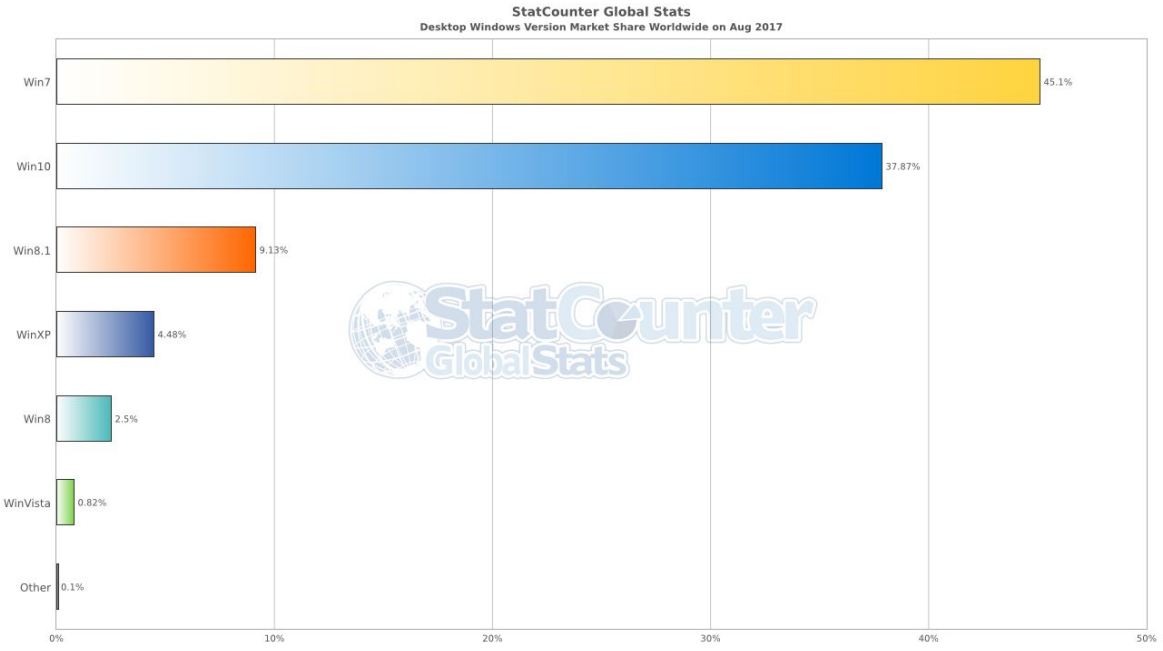
Dataset Metrics

From the dataset, metrics have been derived for three Microsoft products: Windows, Office, and the Internet Explorer and Edge browsers. We have created visualizations aimed to guide policy makers in their investment decisions.

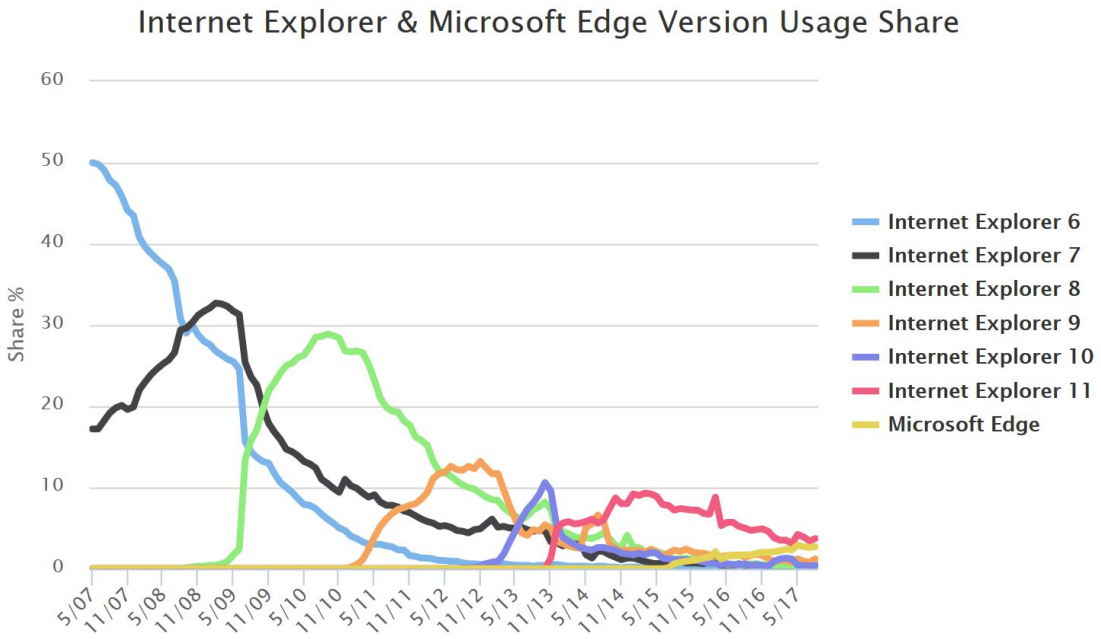
Software version distribution

Users do not always update their software, which means they could be running software with known vulnerabilities. The dataset lists products and their associated version numbers, so a metric that could be derived from the dataset is the amount of vulnerabilities per version of a product. When doing so, it is also important to know how many users are still using older versions.

Below are statistics from StatCounter[8] that show the usage of the products over time. These metrics will help policymakers decide how much budget should be spent on, for example, campaigns to get users to update their software regularly.



[8] Source http://gs.statcounter.com/os-version-market-share/windows/desktop/worldwide



9

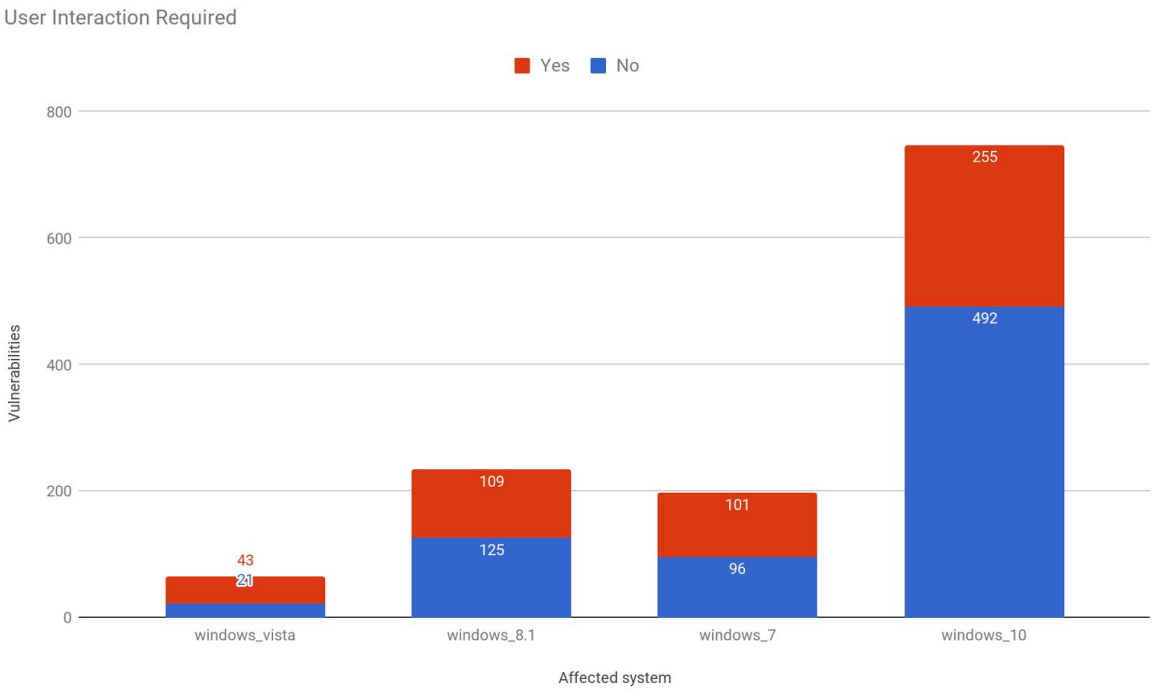
[9] https://www.w3counter.com/trends

User Interaction Required

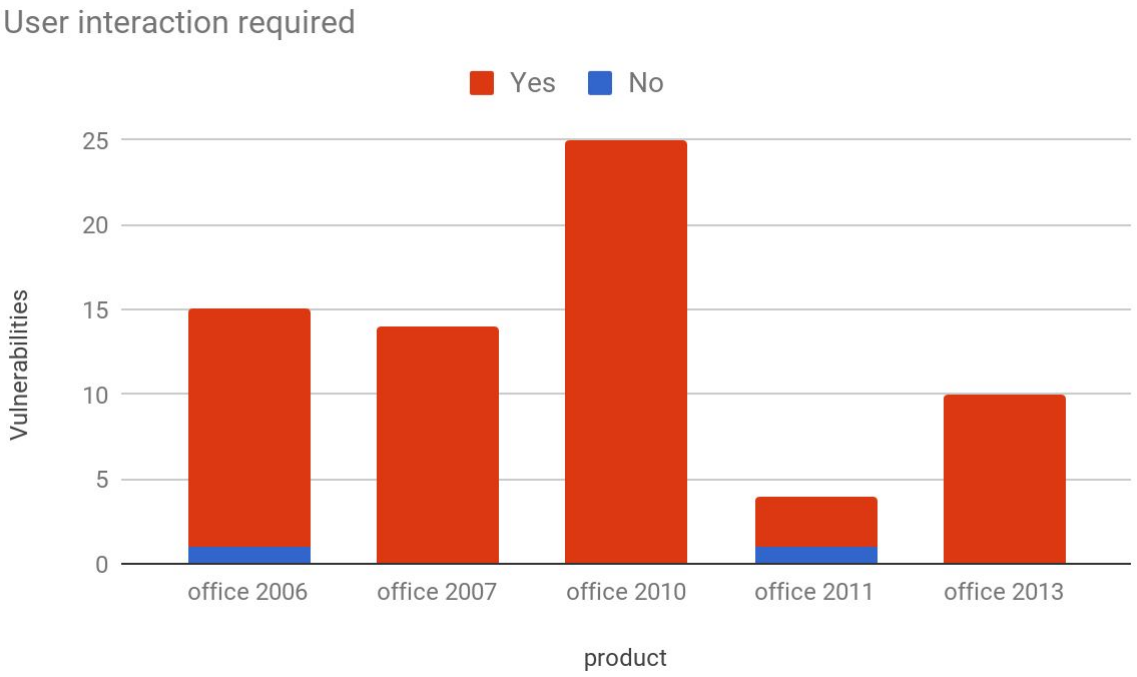
A vulnerability can be misused either with user interaction or without it. An example of user interaction could be opening a malicious document or clicking on a malicious link.

Visualizing whether or not user interaction is required can be useful to find out if Microsoft should launch marketing campaigns on the effects of clicking on unknown objects or building more fail-safes that will proactively protect users from such interactions.

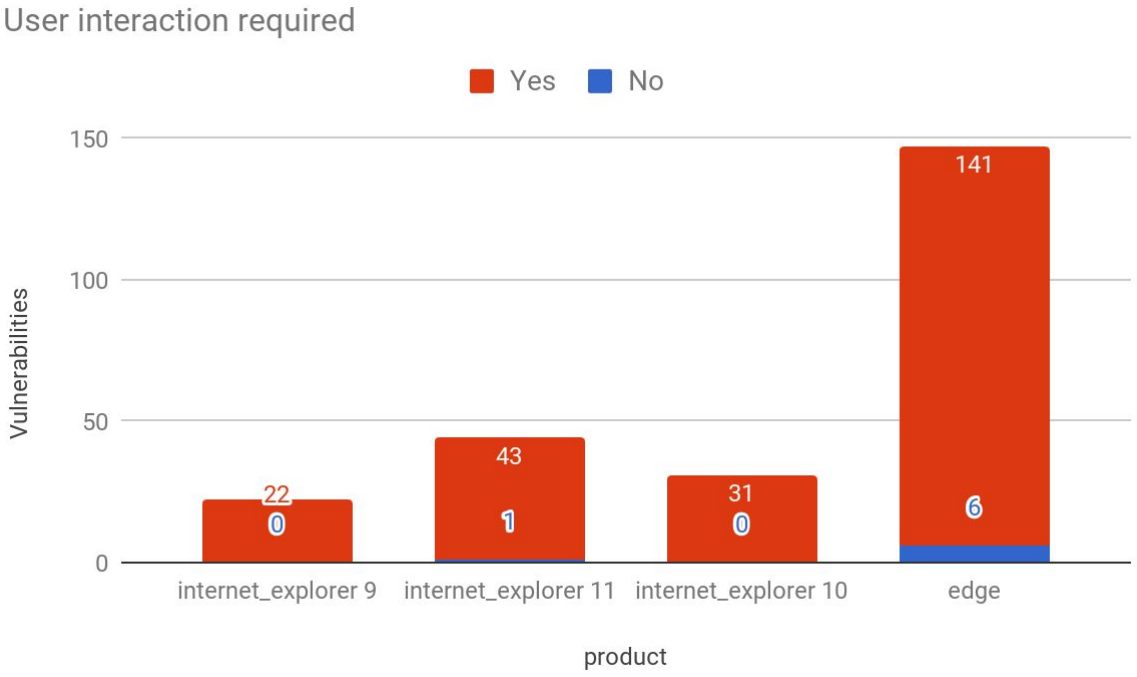
Windows



Office



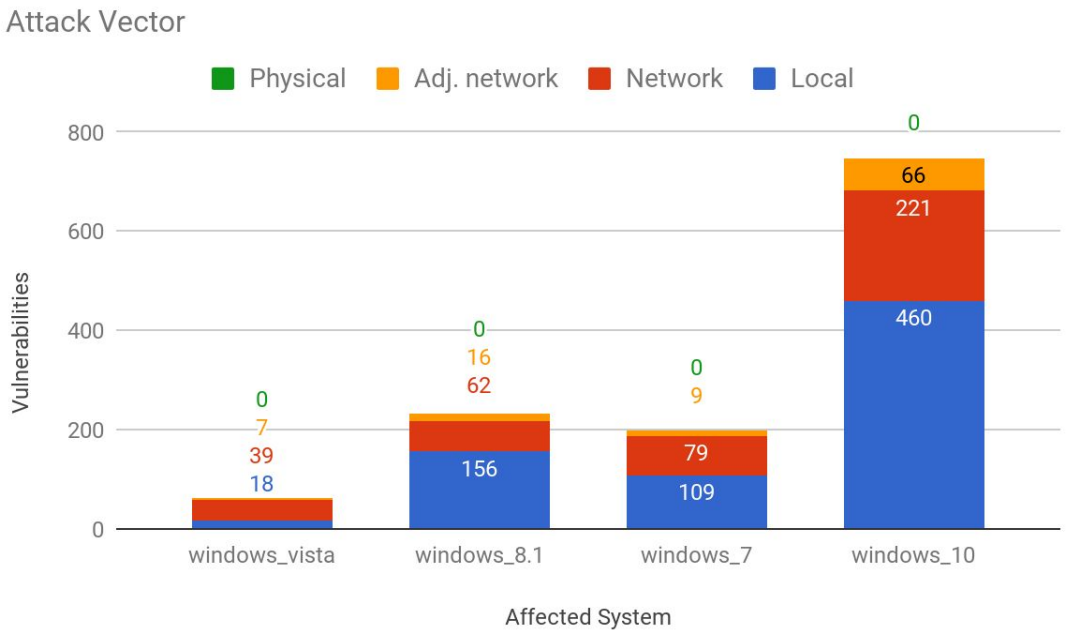
Browsers



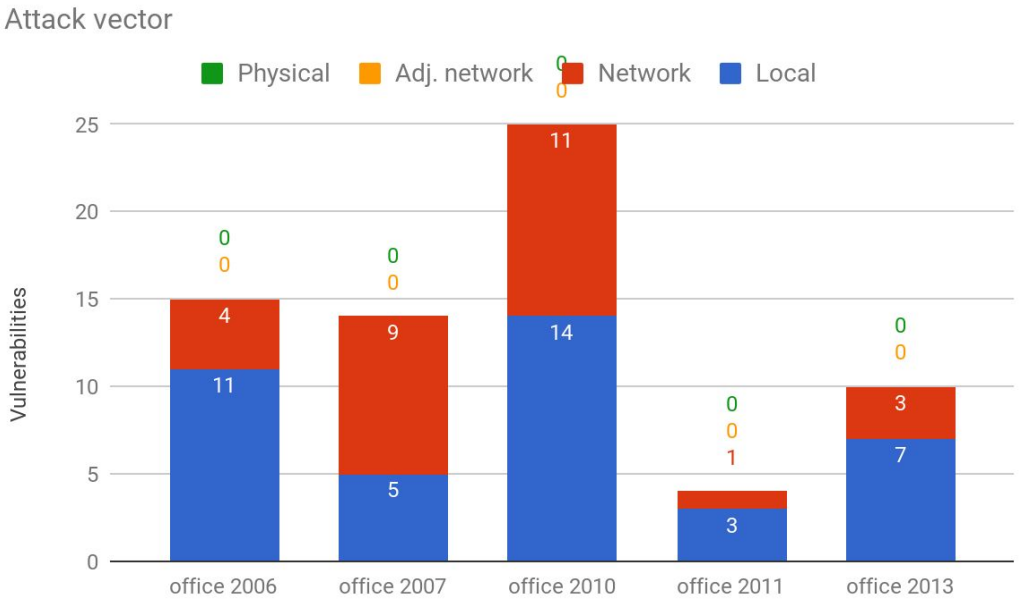
Attack Vector

A vulnerability can be abused over a given attack vector, either local, adjacent network, or network. A vulnerability that can be abused remotely is far more harmful than ones that can only be used locally, because the audience that can use a network exploit is much higher. Therefore, these vulnerabilities should be patched first.

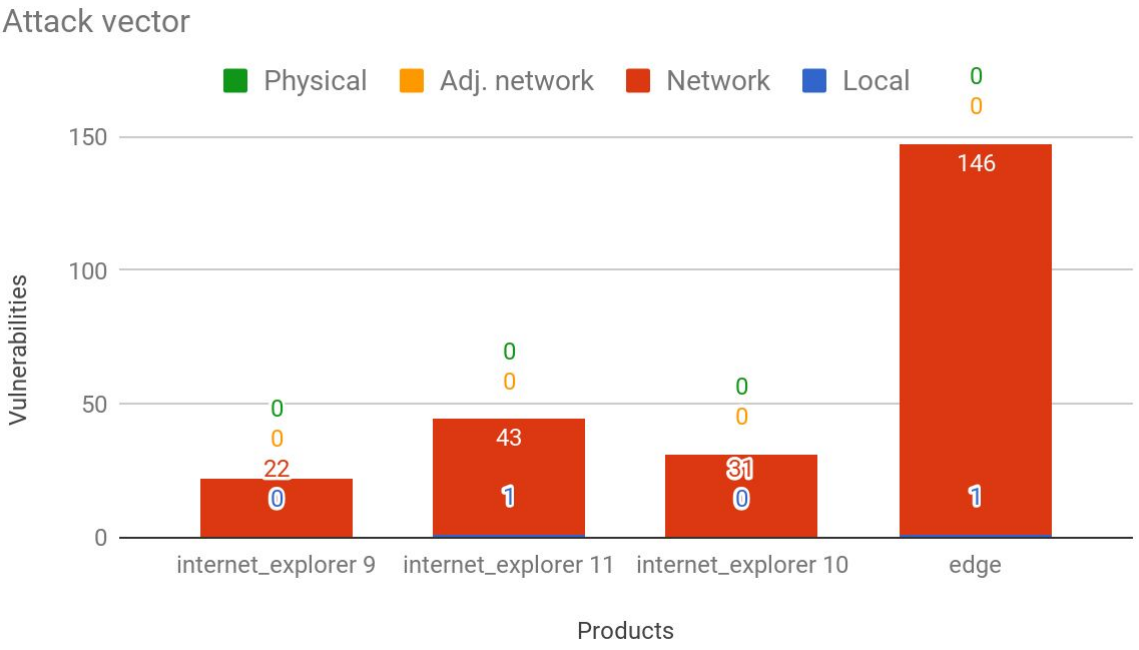
Windows



Office



Browsers


Attack vector
Physical
Adj. network
Network
Local
Vulnerabilities
150
100
50
0
0
0
22
0
0
0
43
1
0
0
31
0
0
0
146
1
internet_explorer 9
internet_explorer 11
internet_explorer 10
edge
Products

Observations

Initial observations that emerged from the developed metrics are listed here for information. These may be the starting point for recommendations for decision makers, although this is considered out of scope.

Windows 7, which is the second most used operating system seems to be less vulnerable to attacks when compared to the most used operating system Windows 10. We can argue that Windows 7 is the stronger version here, all though with further improvements in Windows 10, the usage of 7 will reduce with time.

We see a high number of local and network vulnerabilities in Office 2010. Although all of them require user interactions, these vulnerabilities could become a priority for Microsoft due to the large number of users.

Microsoft Edge has a higher vulnerability count than earlier versions of Internet Explorer, and also requires a high amount of user interaction. It is also the second most largely used browser.

Conclusions

Millions of people worldwide use Microsoft products and expect a safe and secure experience. Investments on making secure systems and keeping these systems secure is a major concern.

In particular, one of the largest dangers today is the widespread use of botnets to infect user systems. It is in Microsoft's best interest to prevent their users from becoming infected by malicious software because they have this responsibility and they will lose profit if systems are affected.

Our research question was as follows: **Suppose we are security policy makers working for Microsoft. How should additional budget be distributed to prevent botnet infections under our users?**

We have used the National Vulnerability Database to develop metrics that may be used to evaluate Microsoft products. By looking at ideal metrics and metrics used in practice today, we have derived metrics that make use of the NVD dataset. Initial observations have been drawn to illustrate how a decision maker might go as to allocating investments.