Prioritisation of Network Security Services

Zia Hayat*
University of Southampton
Communications Laboratory
School of Electronics & Computer Science
Faculty of Engineering
Southampton
Hampshire S017 1BJ
UK
zia.hayat@baesystems.com

Jeff Reeve
University of Southampton
Communications Laboratory
School of Electronics & Computer Science
Faculty of Engineering
Southampton
Hampshire S017 1BJ
UK
jsr@ecs.soton.ac.uk

Chris Boutle
BAE SYSTEMS Integrated System Technologies
Frimley
Camberley
Surrey GU16 7EX
UK
chris.boutle@baesystems.com

*Sponsored by BAE SYSTEMS Integrated System Technologies and EPSRC, UK

Abstract On large networks security administration tasks such as patch management and event log analysis can take many hours and even days to successfully complete even with automated solutions. Currently it is left to the systems administrators' discretion to choose in which order to protect individual devices. In light of the rapidly decreasing time between vulnerabilities being discovered and maliciously exploited by malware, such an arbitrary method introduces an unacceptable level of risk to the security of those devices, which are critical to business processes.

An information risk management approach needs to be adopted to ensure the protection of the network with a high likelihood; this can be achieved through the prioritisation of critical devices. In this introductory paper a generic prioritisation technique for individual devices in a network is described offering a methodical alternative to the current ambiguity of a systems administrators operations. The technique is based upon compromise path analysis, which identifies critical paths in a network from a security viewpoint and is relevant in a wide range of operations from the application of security services to analysing their results. The vulnerability period metric is introduced, as a mechanism to control the risk exposure to individual devices through prioritisation.

Key Words: Computer Network Security, Malware, Information Risk Management, Prioritisation, Compromise Path Analysis & Vulnerability Period.

1 Introduction

In the modern world of always-on (24-7) business needs there is an increasingly inherent reliance of governments, large corporations and the public at large on

Information Technology (IT) networks to carry out integral and critical tasks. The security of such key networks has arisen as a major business and political issue. As described in [1] the damage caused by attacks on IT networks is very significant: network down time can result in disruption to vital business processes; repairing compromised devices can take weeks and consume excessive amounts of people resource; and loss of reputation can be hard to quantify but can potentially be the most significant cost.

It is argued that if the current trend of reducing time scales in exploitation and increasing number of software vulnerabilities continues as reported in [2] then, critical business processes will be exposed to unacceptably high levels of risk, even in the presence of automated security services such as path management as advocated in [3]. Therefore the efficient and effective utilisation of available security resources is required in order to further exploit their potential. This is highlighted in [4] where it is suggested that many organisations fail to gain real benefit from their investments in IT systems. Prioritisation of devices to receive security servicing is an approach, which may be used to enable this, reducing the time frame within which critical assets may be compromised; we call this time frame the Vulnerability Period (VP). The VP for each susceptible device is the time between a vulnerability first being reported to the time at which that device is made secure from such a vulnerability. The VP is defined as follows,

$$VP = P \cdot \tau + K \tag{1}$$

where P is the priority (integer value starting from '1', which is the highest priority) assigned to a given device, τ is the average time taken for a service

to be successfully performed per device, τ is also variable due to differences in: individual services, network latencies and dynamic characteristics of individual devices; K is the time taken for the developers to provide a solution to fix the vulnerability from the time of its discovery.

In [1] Brown et al. describe a pro-active malware susceptibility testing technique (Active Countermeasures), which applies a vaccine (a virus with a NULL payload) to devices on a network. An automated response is sent back from the device under investigation indicating its susceptibility to the virus and if a device is found to be susceptible it is immediately made safe. The scanning of individual devices is based upon a SETI@home-style [5] setup. Using this technique networks are separated into clusters of devices with each cluster being assigned to a given scanner for inspection purposes. However this process provides the service to devices in an arbitrary sequence. From equation 1 it can be seen how prioritisation unlike an arbitrary technique can aid the owners of devices to control the risk exposure to a given device by reducing or increasing its VP accordingly. Without prioritisation the VP can be any time between one and n (number of devices under consideration) times the average time (τ) taken to service a single device.

The use of prioritisation to manage information security risks in IT networks is endorsed in [6] and [7]. In both of these papers the authors identify the requirement for determining the priority and therefore sequence in which individual devices receive security servicing. The question is how to develop a systematic technique for achieving such prioritisation? In [8] the authors introduce Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE) a high-level strategy designed to identify the relative criticality of information and

infrastructure assets. Although this methodology provides a valuable context-driven systematic approach to managing information security risks, it does not directly address the issue of how one would technically achieve prioritisation and therefore control the risk exposure through the length of the VP for individual devices.

The application of security services such as automatic updating of anti-virus patching and signatures, security event log analysis as well as stateful (deep) packet analysis are a few examples of where a prioritisation capability could be employed as part of an overall quality of security service strategy, reducing the impact of potential compromises. For example in the case of deep packet analysis, the excessive overheads associated with scanning network communications as described in [9] means only a subset of the total traffic can be scanned. Therefore a prioritisation technique would prove useful in differentiating between network traffic for analysis purposes, based upon its potential impact if it was to compromise the end-point(s).

Security services may also be significantly degraded if the infrastructure used to facilitate them is compromised and brought offline. This is particularly true if the resources (CPU time and communications bandwidth) of 'ordinary' devices on the network are required to enable such services, which is the case for the SETI@home-style setup used in the Hewlett Packard (HP) Active Countermeasures technique as described previously. It is argued that if a network was to use a security service(s) based upon such an architecture then the service(s) would be severely affected if a worm such as Sasser [10] was to attack it, rapidly limiting available resources. In such a scenario a prioritisation technique would enable the scarce resources available to be focused at those devices perceived to

be of greatest business value.

This paper is organised as follows, section 2 describes a simple prioritisation strategy and highlights its limitations, section 3 details how connectivity in modern IT networks has evolved and how this has impacted the security risks, resulting in our investigation into compromise path analysis to quantify such risks, this is followed by an overview of how networks are modelled in this paper. In section 4 we specify and explain through an example, the algorithm developed to prioritise the order in which individual devices should receive security servicing. A brief overview of preliminary results from testing carried out on a software implementation of the algorithm is provided in section 5, with a summary and potential future work detailed in section 6.

2 A Simple Prioritisation Strategy

In order to develop any sort of information security risk strategy one must derive a security criticality classification system. This involves the identification and valuation of assets following which prioritisation can be derived, based upon the risks to those assets. Traditionally a classification system consists of a number of criticality levels, where individual devices are assigned to one such level. There may be a different classification system for each of the three principle security properties of confidentiality, integrity and availability however, for the purposes of the modelling described in this paper we use only one classification system, which is analogous to that commonly used by many organisations for information security classifications. The criticality levels used in this system are (from high to low criticality): VH, H, M, L and S.

A simple prioritisation strategy for security purposes would be to order the application of security services based upon the criticality level to which a device is assigned, hence all devices with a criticality level of 'VH' would be assigned the highest priority for security service applications. However this may still result in a relatively arbitrary prioritisation of devices if there are a significant number of devices at the same criticality level in a network. This is illustrated in equation 2,

$$P_s = \frac{1}{\eta} \tag{2}$$

where it can be seen that as the total number of devices at the same criticality level in a network increases, the probability of any one device (P_s) receiving the appropriate prioritisation decreases. Therefore the probability of every device (P_a) receiving servicing in the correct order at any given criticality level can be defined as,

$$P_a = \frac{1}{\eta!} \tag{3}$$

Equation 3 further highlights the limitations of an arbitrary servicing strategy when considering all devices at the same criticality level.

From equations 2 and 3 it is clear that in order to successfully prioritise, one requires a differentiating factor(s) between a set of devices at the same criticality level, otherwise the VP for assets will remain undetermined, where large numbers of devices are assigned to the same criticality level. One way of achieving such differentiation is to identify and analyse dynamic risks to individual devices and then prioritise the application of security services based upon these perceived risks. Numerous techniques such as: [11], [12], [13], [14] and [15] exist

for the identification, analysis and ranking (prioritisation) of risks in a network scenario. However a major limitation with all of these techniques is that they assume an exhaustive search of the problem space (i.e. identify individual risk(s) to device(s) and then quantify these based upon the attacker model), which can be vast and complex in the case of security vulnerabilities in modern IT networks.

In order to achieve our goal of pragmatic risk management it was decided to abstract from specific instances of threats, instead concentrating on perceived threats from other devices based upon their criticality level. The classification system used for the purposes of this work assumes that individual devices are assigned to a criticality level based upon the information and services they host. In order to reduce security risks to devices with increasing criticality a number of assumptions have been made for the purposes of this model. Devices of increasing criticality are said to be:

- Accessed by authorised users in whom one has higher degrees of confidence
- More frequently monitored for abnormalities
- Given higher prioritisation for anti-malware purposes

Based upon the aforementioned assumptions the security risk associated with devices of increasing criticality are likely to decrease.

3 Evolving Risks in Evolving Networks

One of the major changes in computing networks over the last few years has been the ability to conveniently form agile business processes through paradigms based upon wireless ad-hoc communications, service oriented architectures and

in the longer term grid (or utility) computing. This has resulted in rapid and complex interconnections between devices previously unimaginable. Such complex interconnections result in open and dynamic connectivity across traditional networking boundaries not only at a personal or home and small office level, but increasingly in the constrained environment of the larger corporate and government network, where sensitive information and services must be protected.

The majority of corporate users today utilise computing devices with constrained communications capabilities for a limited set of well-defined services, however in the near future these devices will be used to deliver much more varied and flexible services using the most functionally and cost effective communications technique available. This can be seen in the desktop computer which is no longer only enabled for communications over the wired corporate backbone, but is increasingly capable of communicating through a multiplicity of interfaces such as Infrared and Bluetooth. In addition to the desktop other devices such as the Personal Digital Assistant (PDA) are also being introduced enabling ubiquitous capabilities through mobile and heterogeneous[†] communications.

Another key driver of the complex interconnectivity between devices will be the enablement of an all Internet Protocol (IP) communications system through the concept of IP Multimedia Subsystem (IMS). IMS is being developed by the telecommunications industry, allowing individual devices such as mobile phones to directly communicate pushing and pulling content and services to and from one another in a distributed fashion unlike the principally client-server internetworking model of the past. An example of how the traditional internetworking environment is evolving is given in figures 1 and 2 respectively. From figure 2 it

[†]Bluetooth, WiFi, General Packet Radio Service (GPRS), Universal Mobile Telecommunications System (UMTS), Ultra Wideband (UWB) and Wireless Broadband (WiMAX).

can be seen that a much more complex and meshed environment is developing, where every device has the potential to become a gateway to external networks. The complexity introduced by such extended connectivity adds to the threat vector in IT networks.

Although the business benefits of using a more a flexible approach to inter-networking are clear the complexities introduced pose significant technical challenges, not least from a security point-of-view. One such challenge is to provide tailored security services according to the diverse needs of individual users and their devices as described by the Jericho forum [16] and their vision of de-perimeterised security solutions. De-perimeterisation can already be seen to be taking place, where distributed or embedded firewalls are utilised on individual devices, supplementing the static and skeletal security model of the monolithic perimeter firewall solution.

3.1 Network Modelling

Throughout this paper we represent network connectivity as a graph $G=\langle D,C\rangle$. $\forall d\in D$, where d denotes a physical device and $\forall$c$\in$C, where c is an inter device compromise connection. We use semantic network modelling as described in [7] to represent compromise network connectivity, where a compromise connection (c) represents the potential to exploit a flaw(s) by one device in another therefore enabling the spread of malware such as viruses and worms. For example if a device hosts a particular web service application, which is subsequently found to have a flaw (e.g. buffer overflow) then all external devices authorised to access the application are then considered to have a compromise connection to the device hosting the vulnerable application. In [7] the author identifies the need to use either directed or un-directed edges with the added concept of attributes

to provide further details on particular protocols (e.g. http, https) being run between devices. In our current model a simplified version of this is used, where all links between devices are undirected and attributes are not considered for simplicity.

A criticism of using such a model driven approach for the description of an IT architecture, is that the model may quickly become out-of-date particularly where mobile and ad-hoc devices are used, which have the ability to create and destroy links dynamically. However there are a number of mature standards such as the Simple Network Management Protocol (SNMP) and Common Information Model (CIM) with corresponding tools such as Cheops-ng [17], Nmap [18] and HPOpenView, which provide semantically rich network topology information in real-time. These standards are capable of providing descriptions in a number of formats, which may be used by techniques such as those described in this paper for subsequent processing and analysis.

Let us first define, $\forall d \in D$:

- The criticality level of d, c_r to be the security criticality associated with the device, this attribute is referenced as $d \cdot c_r$
- The compromise path risk level of d, p_r to be the risk level of the highest threat compromise path associated with the device, this attribute is referenced as $d \cdot p_r$
- The compromise path length of d, h_o to be the length of the compromise path associated with the device, this attribute is referenced as $d \cdot h_o$
- The residual risk of d, r_r to be the residual risk associated with the device, this attribute is referenced as $d \cdot r_r$

- The prioritisation status of d, s_p to be the priority for receiving security servicing associated with the device, this attribute is referenced as $d \cdot s_p$

And, $\forall c \in C$:

- The risk level of c, v_b be the connection risk level associated with the connection, this attribute is referenced as $c \cdot v_b$
- The distinct devices incident to c, e_s be the two devices associated with the connection, this attribute is referenced as $c \cdot e_s$

From the seven attributes of a device four (c_r, p_r, h_o and r_r) are used to identify the relative risk posed to individual devices in the network G. The technique proposed in this paper uses at least one and at most all four of these attributes to prioritise devices. It is assumed that $\forall d \in D$ the administrators of the network G are able to identify the security criticality level c_r based upon a technique such as Her Majesty's Government (HMG) infosec security classifications or OCTAVE [8]. Devices with a higher criticality level c_r have a higher priority, where $c_r \in \{\text{VH, H, M, L, S}\}$.

3.2 Compromise Path Analysis

Although IP networking has the potential to allow ubiquitous connectivity, communications between devices are limited due to routing restrictions imposed by security services (e.g. embedded firewalls). This implies that if an attacker wants to compromise a given device(s) (victim) using a worm for example, then they must launch an attack from a device(s) which is authorised to connect to the victim otherwise the attempt to connect will be rejected. We call such compromise connections compromise paths, where a compromise path consists of one or a series of compromise connections as described in section 3.1. This implies that an attacker has to systematically traverse a number of devices and

overcome various security barriers (defence in depth strategy) such as IP routing rules, traffic analysis and packet inspection to compromise a specific device in a specific manner.

The technique of compromise path analysis as adopted in this work is based upon that developed by QinetiQ for their Domain Based Security (DBSy) methodology [19]. To our knowledge there is no other strategy in the literature similar to compromise path analysis, which explicitly identifies the threat to a device based upon the graph theory concept of reachability. We use security criticalities to bound the search space, by only analysing for compromise connections to devices of equal or lower criticality, as they are deemed to pose the main risk. In the DBSy model, compromise paths are rated according to a technique analogous to table 1, this is part of a HMG infosec standard [20], for quantifying risks when connecting devices of differing security criticalities.

Devices with a criticality of 'S' are not prioritised at all as they are believed to be out of the current administrative authorities control, they are only used as potential attackers in our compromise path analysis technique. Thus the set of devices in a network G which are prioritised (potential victim devices) is: $D_M = \{d : D | d \cdot c_r \neq \text{'S'}\}$. Therefore if for a set of devices, $D_s \subseteq D_M$, one is unable to prioritise between them based upon security criticality levels (c_r) alone (i.e. they have the same criticality level), then the highest compromise path risk (p_r) associated with such devices may be used to distinguish between them. Where, $\forall d \in D_s$, devices with higher compromise path risk levels are given increased priority.

From table 1 it can be seen that a path between a device of 'VH' criticality and one of 'L' criticality has an associated risk level of '4', where a level '1' risk is

Victim \ Attacker	VH	H	M	L	S
VH	1	2	3	4	5
H	-	1	2	3	4
M	-	-	1	2	3
L	-	-	-	1	2
S	-	-	-	-	1

Table 1: Risk levels for connected devices.

the lowest level and a level '5' is the highest risk level. It is also illustrated (i.e. lower half of table is irrelevant) in table 1 that devices are only perceived to be at risk from those of an equal or lower criticality level.

3.3 Extending Compromise Path Risk Analysis

As well as using compromise path risk levels to distinguish between a set of devices as is described in the original method in [19], our technique builds upon this by also using the concepts of compromise path length and residual risk. Therefore if a set of devices $D_H \subseteq D_M$, have the same criticality and compromise path risk levels, then compromise path length (number of hops between victim and attacker device) is used to distinguish between them. Where, $\forall d \in D_H$, devices with lower compromise path lengths are given higher priority. This is due to the fact that the attacker device ($d \in D$) is closer to the victim device ($d \in D_H$) resulting in less effort to compromise the victim on behalf of the attacker. We further extend our risk analysis by introducing the concept of residual risk, therefore if a set of devices, $D_R \subseteq D_M$, have the same criticality level (c_r), compromise path risk level (p_r) and compromise path length (h_o), then residual risk (r_r) is used to distinguish between them. Where, $\forall d \in D_R$, devices with higher residual risk are given higher priority. The residual risk (r_r) of a device $d \in D_R$ is calculated as,

$$d \cdot r_r = \frac{(\Sigma c \cdot v_b)^2}{\zeta} \tag{4}$$

where ζ is the number of directly connected devices which have an equal or lower criticality level than the device (d), and v_b is the connection risk level for such connections ($c \in C$) and is calculated by comparing the difference in criticality levels according to table 1 of the two devices in $c \cdot e_s$.

We quantify the risk to a device by assigning the four metrics described previously to attributes of each potential victim device $d \in D_M$ in the network G. The attributes corresponding to each metric and in order of precedence are:

1. c_r - Criticality level
2. p_r - Compromise path risk level
3. h_o - Compromise path length
4. r_r - Residual risk factor

Therefore a metric of lower precedence is only used if a higher order metric is unable to provide prioritisation for a given set of devices.

If for a device $d \in D_M$ both p_r and h_o attributes have a value of '0' this indicates no compromise path exists for d. Otherwise if a compromise path exists p_r takes a value, $1 \leq p_r \leq 5$, where '1' represents the lowest risk and '5' the highest; and h_o takes a value, $h_o > 0$. The r_r attribute is calculated to distinguish between devices, which cannot be prioritised using a combination of c_r, p_r, and h_o attributes, this is specified in the algorithm in section 4, where $r_r \geq 0$.

4 Specification of Algorithm

For the purposes of the algorithm to be described, we will define a number of terms to aid in its understanding.

Definition 1: Let ${D_M}^{(x)} \subseteq D_M$ denote the set of devices with a security criticality of x.

Definition 2: $\forall d \in D_M$, let $p(d)$ calculate the highest risk compromise path of device d, calculated according to table 1 using a constrained[‡] Depth First Search (DFS). If two or more compromise paths of identical risk level exist for one device then $p(d)$ chooses the one with the smallest length, choosing any one if more than one has the same length as well as risk level. Formally $p : D_M \rightarrow \{1, 2, 3, 4, 5\}$.

Definition 3: $\forall d \in D_M$, let $h(d)$ calculate the length of the highest risk compromise path to d. Formally $h : D_M \rightarrow \{n : \mathbb{N} | n > 0\}$, where $\mathbb{N}$ is the set of natural numbers.

Definition 4: $\forall d \in D_M$, let $r(d)$ calculate the residual risk (r_r) of the device d calculated according to equation 4. Formally $r : D_M \rightarrow \{n : \mathbb{R} | n \geq 0\}$, where $\mathbb{R}$ is the set of real numbers.

Our consolidated prioritisation technique is recursive and the algorithm is:

1. $\forall d \in D_M$ set $d \cdot p_r = 0$, $d \cdot h_o = 0$, $d \cdot r_r = 0$ and $d \cdot s_p = 0$
2. **foreach** x:=VH:L let $D_S := {D_M}^{(x)} \subseteq D_M$

[‡]Although beyond the scope of this paper it must be noted that a number of limitations have been imposed upon the search algorithm to ensure it is computationally feasible, whilst providing a comprehensive analysis from a security point-of-view.

3. **if** $|D_S| > 1$ **then**

4. $\forall d \in D_S$, let $d \cdot p_r := p(d)$ and $d \cdot h_o := h(d)$

5. $D_H := \emptyset, n := |D_S|$

6. **for** $t := 1 : n$

7. **for** $i := 1 : n$

8. **if** $((d_t, d_i \in D_S) \wedge (i \neq t) \wedge (d_t \cdot p_r = di \cdot p_r) \wedge (d_t, d_i \notin D_H))$ **then**

9. $D_H{}' := D_H \cup (d_t \wedge d_i)$

10. **end**

11. **end**

12. **end**

13. **if**$(D_H = \emptyset)$ **then**

14. $\forall d \in D_S$, set priority attribute $d \cdot s_p$ giving increased priority to devices with higher $d \cdot p_r$ values

15. **end**

16. **else**

17. $D_R := \emptyset, D_\delta := \emptyset n := |D_H|$

18. **for** $t := 1 : n$

19. **for** $i := 1 : n$

20. **if** $((d_t, d_i \in D_H) \wedge (i \neq t) \wedge (d_t \cdot h_o = di \cdot h_o) \wedge (d_t, d_i \notin D_R))$ **then**

21. $D_R{}' := D_R \cup (d_t \wedge d_i)$

22. **end**

23. **end**

24. **end**

25. **if**$(D_R = \emptyset)$ **then**

26. $\forall d \in D_H$, set priority attribute $d \cdot s_p$ giving increased priority to devices (d) with lower $d \cdot h_o$ values

27. $\forall d \notin D_H \wedge d \in D_S$, reassign priorities for such devices accordingly

28. **end**

29. **else**

30. $D_\delta := D_H \setminus D_R$

31. **if** $(D_\delta \neq \emptyset)$

32. $\forall d \in D_\delta$, set priority attribute $d \cdot s_p$ giving increased priority to devices (d) with lower $d \cdot h_o$ values

33. **end**

34. $\forall d \in D_R$, set $d \cdot r_r = r(d)$

35. $\forall d \in D_R$, set priority attribute $d \cdot s_p$ giving increased priority to devices (d) with higher $d \cdot r_r$ values

36. **if** $\exists d \in D_R$, $d \cdot r_r$ attribute values are identical **then**

37. assign equal priority to such devices

38. **end**

39. $\forall d \notin D_R \wedge d \in D_S$, reassign priorities accordingly

40. **end**

41. **end**

42. **end**

43. **else if** $|D_S| \leq 1$ **then**

44. do nothing

45. **end**

46. **end**

47. $\forall d \in D_M$ reassign priority values for devices according to the c_r attribute values

4.1 Algorithm Execution

In order to illustrate the prioritisation strategy and the individual attribute values calculated by the algorithm when executed we have analysed the network given in figure 2. It is assumed that the file, mail and web servers are remotely accessible using the Remote Desktop Protocol by administrator accounts, where currently those accounts are active on devices 'A', 'E' and 'B' as depicted in figure 3. Numerous (i.e. B-D, K-I etc.) other devices are also connected using this protocol as can be seen from figure 3. However it is assumed that a flaw has been identified in a common application of this protocol, the connectivity due to this is represented as 'Compromise 1' edges in the graph in figure 3, where the criticality level associated with a device is given in brackets. Another flaw allowing a Trojan Horse worm is then assumed to have been discovered affecting certain versions of both user and server operating systems; this is represented by the 'Compromise 2' connections (edges) in figure 3. If considered simultaneously the connectivity due to the aforementioned compromises is as depicted in figure 3, this is commonly the case with sophisticated worms which

utilise numerous flaws or even if one decided to analyse a network for currently popular vulnerabilities.

The connections between devices B-Mail Server, F-D etc. depicted in figure 3 illustrate the fact that these devices are using the vulnerable version of the application running the Remote Desktop Protocol, and more importantly allow one another to utilise this application (i.e. firewall on device 'Mail Server' is configured to allow device 'B' to connect and utilise this service). Thus in the case of an attack if device 'B' is compromised it will be able to compromise the device 'Mail Server' even if 'Mail Server' has an embedded firewall enabled.

Table 2 details the values for each attribute of each device after the execution of the algorithm specified in section 4, when considering only 'Compromise 1'. Table 3 gives details when both compromises 'Compromise 1' and 'Compromise 2' are considered simultaneously.

Device	c_r	p_r	h_o	r_r	s_p
File Server	VH	5	4	0.0	3
Mail Server	VH	5	3	18.0	1
Web Server	VH	5	3	4.0	2
A	H	1	1	0.0	8
B	L	1	1	0.0	13
C	H	4	2	12.5	4
D	M	3	3	0.0	10
E	H	4	2	4.5	5
F	H	4	4	0.0	7
G	L	2	1	4.0	12
I	H	4	2	4.0	6
J	L	2	1	4.5	11
K	M	3	1	0.0	9

Table 2: Attribute values after execution when considering 'Compromise 1'.

Device	c_r	p_r	h_o	r_r	s_p
File Server	VH	5	4	0.0	3
Mail Server	VH	5	3	0.0	2
Web Server	VH	5	1	0.0	1
A	H	1	1	0.0	8
B	L	1	1	0.0	13
C	H	4	2	21.3	4
D	M	3	3	0.0	10
E	H	4	2	4.5	5
F	H	4	4	0.0	7
G	L	2	1	4.0	12
I	H	4	2	4.0	6
J	L	2	1	4.5	11
K	M	3	1	0.0	9

Table 3: Attribute values after execution when simultaneously considering 'Compromise 1' and 'Compromise 2'.

5 Preliminary Experiment

The algorithm described in section 4 has been developed in a C#.NET software version called **pfcca**. The **pfcca** software requires an Extensible Markup Language (XML) file as input, which adheres to an XML Schema Definition describing devices and their associated compromise connectivity. However due to the time consuming nature of developing such XML input files by hand it was decided that automatic compromise connection network topology generation was required for preliminary testing. We developed a simulator named **generator**, which emulates the potential for compromise connections between devices using a theoretical attack model based upon that of the Blaster [21] worm and a localised connectivity strategy[§] as described in [23]. The details of the simulator are beyond the scope of this paper, however it is worth noting that well known power-law strategies [24], [25], [26], [27] for IP network connectivity form the basis of our method, with a subset of this connectivity chosen to represent the potential for exploitation (compromise connectivity) using our

[§]Much more likely to infect IP addresses close to its own address, which was a strategy employed by the Code Red II [22] worm.

theoretical attack. The **generator** application randomly assigns a criticality level to each device in a network, assigning higher criticality levels to critical nodes such as machines with administrator privileges for servers.

The theoretical attack emulated in the **generator** software is assumed to affect the majority of operating systems in use, allowing a remote attacker to gain unauthorised privileges using an open Transmission Control Protocol (TCP) port to automatically replicate and even compromise the confidentiality and integrity of information on susceptible systems. It is believed that this is a realistically achievable attack[¶] due to the upsurge in use of Commercial Off-The-Shelf (COTS) computing components, which have very similar fundamental structures and therefore limited defence-in-depth. For example the particular bug exploited in the Blaster worm affected default installations of Windows NT 4.0, Windows 2000, Windows XP as well as Windows 2003 Server. Other forms of malware based upon scripting languages such as Visual Basic (VB) and Java Server Pages (JSP) take advantage of popular functionality such as Microsoft Office and the Java Virtual Machine to propagate and cause a major impact. This was illustrated with the spread of the Melissa [28] macro virus, which used VB and various common Microsoft Office components to infect and spread.

In order to visualise the networks (graphs) produced by the **generator** software a **MATLAB M-file** is also created by the simulator, which contains the adjacency matrix of the graph (G). The **gplot** function in **MATLAB** is then used to draw the network and provide a visualisation of the networks described in the corresponding XML file.

[¶]Even in the presence of distributed firewalls as the attacker may still be able to attack a system indirectly through intermediate systems which may have trust relationships.

To test the effectiveness of our strategy we ran 6 tests over **pfcca** with each test consisting of 100 networks (created by **generator**) of random sizes (ranging from 10 to 1000 nodes) and compromise connectivity. The aim of the test was to identify what percentage of devices in a network where actually prioritised by the four attributes of: criticality (c_r), compromise path risk level (p_r), compromise path length (h_o) and residual risk (r_r). The results are given in table 4.

Metric	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**	**Mean**	**S. Dev**
c_r (%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p_r (%)	11.92	12.00	11.53	12.33	11.39	12.64	11.89	0.47
h_o (%)	36.39	36.54	36.41	36.29	37.21	36.37	36.79	0.34
r_r (%)	50.99	50.87	50.91	50.74	50.48	50.05	50.45	0.35
None (%)	0.70	0.59	1.15	0.64	0.92	0.94	0.87	0.22

Table 4: Comparison of metric performance for prioritisation purposes.

5.1 Experiment Analysis

From the results in table 4 it can be seen that approximately 99.13% of prioritisation decisions for the six tests (600 networks in total) were based upon one of the four attributes chosen. In comparison 0.87% of decisions could not be made using any of the four attributes. This highlights the usefulness of our consolidated prioritisation technique for successfully prioritising devices in a network. However this does not discount the possibility for further improvements or more efficient ways in which prioritisation could be achieved. The c_r attribute produced no distinguishing results, this confirms our belief that criticality levels alone are not a suitable distinguishing metric for prioritisation in large networks (or where the number of devices is greater than the number of criticality levels) as discussed in section 2.

6 Conclusions & Future Work

It is believed that prioritisation of network security services is much needed in order to successfully protect critical IT assets, particularly as many networks are allowing more diverse and extended connectivity, which is resulting in increased security risks. It has been shown that security classifications alone, do not provide the required level of granularity upon which to base prioritisation decisions in large dynamic networks. Therefore we have developed a strategy based upon compromise path analysis to differentiate between devices, which are assigned to the same high-level static criticality level. Our technique considers dynamic risks from connected devices and quantifies the highest risk posed to any one device.

There are two cases in which even a prioritisation technique would be of limited use, firstly if an attacker has complete knowledge of the prioritisation technique as well as the criticality levels assigned to individual devices then they may target specific devices, to cause maximum impact. Related to this is the fact that an attacker may employ traffic analysis techniques to identify in which order and to what degree individual devices (or network segments) receive security services, thus potentially providing valuable information on the criticality of specific devices (or a group). Secondly it is acknowledged that although our prioritisation technique may limit the impact from malware such as CodeRed, Blaster and even Mydoom [29] it would have limited success in protecting against an extremely sophisticated theoretical attack known as a Flash worm as described in [23]. A Flash worm is able to accurately pre-compute the network addresses of susceptible devices, enabling it to spread through the entire Internet in seconds.

From preliminary testing the dynamic risk analysis technique for prioritisation described in this paper has delivered promising results, distinguishing between the vast majority (99.13%) of devices. Initial testing of the technique for large-scale real-world network topologies has also delivered encouraging results, however if this is to be done comprehensively then one would need to collate the necessary network topology semantics information and analyse[‖] this using a tool such as Nmap to identify potential compromise connectivity between devices. The development of an automated technique to achieve this is proposed as future work. Further work is also required in order to ascertain the effectiveness of the technique for different network topologies and criticality systems with varying levels. It is hoped that the VP can be used as a standard metric to compare the performance of differing techniques for the reduction of risk to critical assets in a network.

7 Acknowledgements

The authors would like to thank Robert Johnston and Tim Parsons for their invaluable contribution to this work.

[‖] Firewall rules applied by individual devices and network edge filters at any one time.

List of Figures

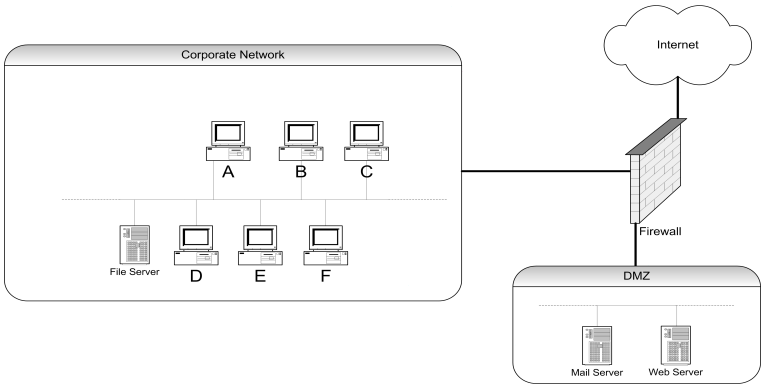

Figure 1: Traditional networking scenario.

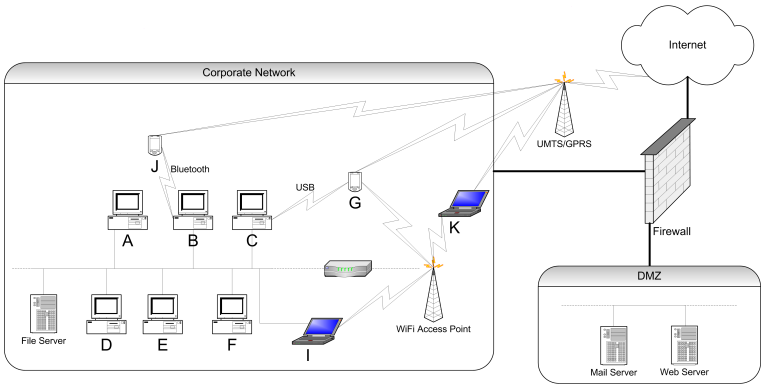

Figure 2: Evolving networking scenario.

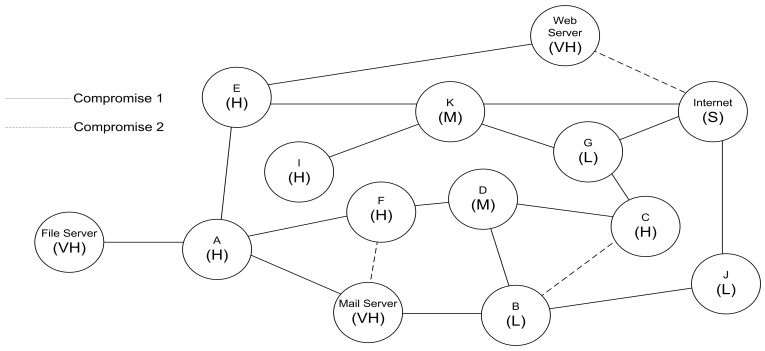

Figure 3: Compromise connectivity due to two vulnerabilities for the network given in figure 2.

References

[1] R. Brown, J. Griffin, A. Norman and R. Smith, "Why HP did not get Blastered", *HP Laboratories Technical Report HPL-2004-188*, Bristol, UK, 2004.

[2] CERT/CC, Available Online: http://www.cert.org/stats/cert_stats.html, last accessed on 17^{th} October 2005.

[3] R. Dacey, "Effective Patch Management is Critical to Mitigating Software Vulnerabilities", Testimony before the Subcommittee on Technology, Information Policy, Intergovernmental Relations, and the Census, House Committee on Government Reform, United States General Accounting Office, Available Online: http://www.iwar.org.uk/comsec/resources/worm-virus-defense/GAO-final-testimony.pdf, last accessed on 17^{th} October 2005.

[4] T. Jennings, "Developing the Role of the CIO", *Information Economics Journal*, pp11-13, London, UK, 2005.

[5] SETI@home, Available Online: http://www.setiathome.ssl.berkeley.edu/, last accessed on 17^{th} October 2005.

[6] L. Rogers and J. Allen, "Securing Information Assets: Security Knowledge in Practice", *CrossTalk Defense Software Engineering Journal*, US Air Force, Available Online: http://www.stsc.hill.af.mil/crosstalk/2002/11/rogers.html, last accessed on 17^{th} October 2005.

[7] B. Monahan, "Infrastructure Security Modelling for Utility Computing", *HP Laboratories Technical Report HPL-2005-04, Bristol*, UK, 2005.

[8] C. Alberts and A. Dorofee, "An Introduction to the OCTAVE Method", Software Engineering Institute Carnegie Mellon University, Available On-

line: http://www.cert.org/octave/methodintro.html#chars, last accessed on 17^{th} October 2005.

[9] Network Appliance Inc., "Antivirus Scanning Best Practices Guide", *Network Appliance Inc. Technical Report TR 3107*, Sunnyvale, California, USA, 2005.

[10] T. Nakayama and F. Ladley, "W32.Sasser.Worm", *Symantec Corporation Security Response*, Available Online: http://securityresponse.symantec.com/avcenter/venc/data/w32.sasser.worm.html, last accessed on 6^{th} February 2006.

[11] J. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla and A. Murukan, "Improving Web Application Security: Threats and Countermeasures", Microsoft Corporation Technical Library, Available Online: http://msdn.microsoft.com/library/default.asp?url=/library/en−us/dnnetsec/html/ThreatCounter.asp, last accessed on 17^{th} October 2005.

[12] B. Schneier, "Attack Trees: Modelling Security Threats", *Dr. Dobb's Journal*, Available Online: http://www.schneier.com/paper-attacktrees-ddj-ft.html, last accessed on 17^{th} October 2005.

[13] C. Salter, O. Saydjari, B. Schneier and J. Wallner, "Toward a Secure System Engineering Methodology", *In Proceedings of New Security Paradigms Workshop*, Charlotsville, Virginia, USA, 1998.

[14] J. Surdu, J. Hill, R. Dodge, S. Lathrop and C. Carver, "Military Academy Attack/Defense Network Simulation", *Advanced Simulation Technology Conference: Symposium on Military*, Government, and Aerospace Simulation, Orlando, Florida, USA, 2003.

[15] A. Moore, R. Ellison and R. Linger, "Attack Modeling for Information Security and Survivability", *Carnegie Mellon University Technical Note CMU/SEI-2001-TN-001*, Pittsburgh, Pennsylvania, USA, 2001.

[16] N. Bleech, "Visioning White Paper, What is Jericho Forum?", Available Online: http://www.opengroup.org/projects/jericho/uploads/40/6809/vision_wp.pdf, last accessed on 17^{th} October 2005.

[17] Cheops, Available Online: http://cheops-ng.sourceforge.net/, last accessed on 17^{th} October 2005.

[18] Nmap, Available Online: http://www.insecure.org/nmap/, last accessed on 17^{th} October 2005.

[19] K. Hughes and S. Wiseman, "Analysis of information security risks: Policy for protection through to implementation", 4^{th} *European Conference on Information Warfare and Security*, Glamorgan, UK, 2005.

[20] Ministry of Defence, Her Majestys Government Infosec Standard No. 3, London, UK, 2001.

[21] D. Knowles, F. Perriot and P. Szor, "W32.Blaster.Worm", *Symantec Corporation Security Response*, Available Online: http://securityresponse.symantec.com/avcenter/venc/data/w32.blaster.worm.html, last accessed on 6^{th} February 2006.

[22] P. Szor and E. Chien, "CodeRed II", *Symantec Corporation Security Response*, Available Online: http://securityresponse.symantec.com/avcenter/venc/data/codered.ii.html, last accessed on 6^{th} February 2006.

[23] S. Staniford, V. Paxson and N. Weaver, "How to 0wn the internet in your spare time", *Proceedings of the 11th USENIX Security Symposium*, San Francisco, California, USA, 2002.

[24] J. Spencer and L. Sacks, "Modelling IP Network Topologies by Emulating Network Development Processes", *IEEE Softcom*, Split, Croatia, 2002.

[25] W. Aiello, F. Chung and L. Lu,"A random graph model for massive graph", *ACM Symposium on Theory of Computing*, pp171-180, Portland, Oregon, USA, 2000.

[26] BRITE, Available Online: http://www.cs.bu.edu/BRITE/, last accessed on 17^{th} October 2005.

[27] C. Palmer and J. Steffan,"Generating Network Topologies That Obey Power Laws", *Proceedings of the Global Internet Symposium*, San Francisco, California, USA, 2000.

[28] K. Tocheva, M. Hypponen and S. Rautiainen, "Melissa", *F-Secure Corporation Computer Virus Information*, Available Online: http://www.f-secure.com/v-descs/melissa.shtml, last accessed on 6^{th} February 2006.

[29] E. Carrera and G. Erdelyi, "Mydoom", *F-Secure Corporation Computer Virus Information*, Available Online: http://www.f-secure.com/v-descs/novarg.shtml, last accessed on 6^{th} February 2006.