Estimates of success rates of remote arbitrary code execution attacks

Teodor Sommestad, Hannes Holm, Mathias Ekstedt

Industrial Information and Control Systems
Royal Institute of Technology

Purpose: To identify the importance of the factors that influence the success rate of remote arbitrary code execution attacks. In other words, attacks which use software vulnerabilities to execute the attacker's own code on targeted machines. Both attacks against servers and attacks against clients are studied.

Design/methodology/approach: The success rates of attacks are assessed for 24 scenarios: 16 scenarios for server-side attacks and 8 for client-side attacks. The assessment is made through domain experts and is synthesized using Cooke's classical method, an established method for weighting experts' judgments. The variables included in the study were selected based on the literature, a pilot study, and interviews with domain experts.

Findings: Depending on the scenario in question, the expected success rate varies between 15 and 67 percent for server-side attacks and between 43 and 67 percent for client-side attacks. Based on these scenarios, the influence of different protective measures is identified.

Practical implications: The results of this study offer guidance to decision-makers on how to best secure their assets against remote code execution attacks. These results also indicate the overall risk posed by this type of attack.

Originality/value: Attacks that use software vulnerabilities to execute code on targeted machines are common and pose a serious risk to most enterprises. However, there are no quantitative data on how difficult such attacks are to execute or on how effective security measures are against them. This study provides such data using a structured technique to combine expert judgments.

Keywords: Remote code exploits, Buffer overflows, Software vulnerabilities, Expert judgment

1 Introduction

The presence of software vulnerabilities in information systems is an important source of risk. Software vulnerabilities can be exploited by adversaries to gain access to sensitive information, to abuse functionality or to consume other system resources. In some cases, it is possible to remove a vulnerability by applying a software patch. In other cases, this type of removal is not possible, either because the vendor has not issued such a patch or because the vendor and the public are unaware of the vulnerability's existence. Also, in many cases, the cost or risk associated with applying a patch (e.g., the service being unavailable during the patching process) hinders the management from applying the patch in a timely fashion.

Software vulnerabilities that can be used to obtain remote control over a machine belong to the most severe examples. Such vulnerabilities are typically exploited by injecting malicious instructions into the memory space of the software that is running on the targeted machine and passes control of the system to the attacker. They are collectively called "arbitrary code vulnerabilities" and include buffer overflow vulnerabilities, dangling pointer references, insecure use of format strings, and integer errors (Younan, 2008).

The risk that an organization faces when such vulnerabilities are present in one of their systems is contingent on the probability that the vulnerabilities can be successfully exploited in practice. Some vulnerabilities are by nature more difficult to exploit than others, and it is possible to apply a number of security measures that makes exploitation more difficult (Younan, 2008).

Because the risk that an organization faces is highly dependent on the likelihood of successful exploitation, data regarding this aspect are very valuable when performing risk analysis, e.g., of a specific vulnerability or when using attack graph approaches such as (Patsos et al., 2010; Sommestad et al., 2010; Sawilla and Xinming Ou, 2008; Homer et al., 2010). However, data on the likelihood of successful exploitation are difficult to obtain because there are many relevant factors for the success of the exploitation. To generalize from observations would require tests on representative samples of vulnerabilities in different environments, with various security measures in place, and involving attackers who are representative of some category of adversary. Thus, it is immensely expensive to gain sound results through experiments, and as a consequence, they are rarely performed. The few experiments that have been performed on the subject have successfully demonstrated technical limitations of measures used in isolation, but they have not reported the difficulty of exceeding these limitations in practice. For example, Shacham et al. (2004) tested the effectiveness of address space layout randomization under certain conditions but do not show how often these conditions apply in practice. Wilander and Kamkar (2003) performed tests of a few protective measures against buffer overflows of different forms. However, without data on the attack forms used in practice, it is difficult to derive useful success rates from this data. Many of the tests that have been performed are of low relevance to practitioners (e.g., network administrators) because they evaluate defense mechanisms that are very difficult to implement, for example, because they are not supported by common operating systems.

Expert judgment is often used when quantitative data are difficult to obtain from experimental studies or by other means. Expert judgment, for example, has been used to assess the importance of attributes that are related to critical infrastructure risks (Cooke and Goossens, 2004) and to quantify parameters in security risk models (Ryan et al., 2010). This paper describes a study in which expert judgment was used to quantify the success rate of remote arbitrary code execution attacks in 24 different attack scenarios.

An important issue when eliciting expert judgment is that of bias. In other words, experts are prone to various types of bias, e.g., relating to their background. This study synthesizes the judgment of 21 domain experts using an established performance-based method known as Cooke's classical method (Cooke, 1991). This method assigns weights to domain experts' judgments based on their ability to estimate the true value for a number of seed questions, that is, questions related to the subject matter and for which the true answer is known. These seed questions are used to identify experts who are suitable to answer the questions of interest, i.e., experts who have both the relevant background knowledge and the ability to express their knowledge quantitatively. Seed questions in this study were designed to find experts that are suitable for estimating the success rate of remote arbitrary code exploits. The experts' performance on these seed questions are used to weight their assessments of the 24 attack scenarios.

The 21 domain experts assessed 16 scenarios related to server-side attacks and 8 scenarios related to client-side attacks. These scenarios are used to analyze the effectiveness of the various defense mechanisms that have value for network administrators or decision-makers in security issues. The uncertainty of these estimates is also described. Both the research method used and the variables included in the scenarios have been previously tested in a pilot study (Holm et al, 2011).

2 Attack scenarios

This study quantifies the probability that remote arbitrary code execution attacks will succeed given that they are executed. Many variables influence whether such an attack succeeds or not. The presence of a software vulnerability that enables the execution of arbitrary code is a necessary condition (e.g., a buffer overflow vulnerability (Cowan et al., 2003)). Some vulnerabilities are only exploitable under certain conditions. Two variables that are often used to describe when a software vulnerability can be exploited are (1) whether the vulnerability can be exploited remotely or locally and (2) whether the attacker would need to bypass some authentication mechanism before the vulnerability can be exploited (Mell et al., 2007).

Furthermore, countermeasures against code execution attacks can be deployed both on a network and a machine level. Deep-packet-inspection firewalls and filtering proxies are two network-based measures that can prevent the executable code from reaching its target (Scarfone and Mell, 2007). Measures that are deployed on a machine level include (Younan, 2008) non-executable memory protection (NX), which makes certain parts of memory impossible to use in executing code, guard page-based countermeasures that terminate programs that access certain parts of memory, execution monitors that execute programs in a "sandbox" or that search for anomalies in execution, address space layout randomizations (ASLR), which obfuscate the memory for the attacker, and instruction set randomizations that encrypt the program instructions so that attackers cannot insert their own instructions without the decryption key.

All of these protective measures have multiple implementations and variants that are available, for example, for different operating system platforms. However, for a variety of reasons, they are not all used in practice. Because the aim of this research is to construct a model that is useful for enterprise decision makers, such as network administrators, the focus is placed on variables that are common in practice. The list of chosen variables was assessed by using the following: 1) literature studies, 2) a pilot study (Holm et al, 2011), and 3) three interviews with respondents who had significant practical experience from arbitrary code attacks.

The chosen variables include the protection mechanisms *NX* and *ASLR,* which are straightforward to turn on or off in commonly used operating systems. Deep-packet-inspection firewalls (*DPI*) and filtering proxies (*Proxy*) are also included as variables. The latter two are also common in today's enterprise environments. Additionally, in server-side attacks, it is significant if the attacker can authenticate himself as a legitimate user (Scarfone and Mell, 2007). Because the existence of authentication mechanisms is something that can be influenced in practice, it is included as the variable *AccessControl* in the attack scenarios. *Connectivity* and *CraftResponse* can be seen as necessary (but not sufficient) conditions for a remote attack. If *Connectivity* is true, then the attacker can connect to the service that is to be attacked; if *CraftResponse* is true, then the attacker can create data that are fed to the client. In practice, the presence of a high-severity vulnerability (*Vulnerable*) is also a necessary condition for remote code execution.

Naturally, the attacker's competence and resources are also of importance to the probability of successful remote code execution. Such attacker properties are kept constant for all of the studied attack scenarios (cf. Table 1). In other words, the attacker is a professional penetration tester who has access to open and commercially available tools and has one week of time to prepare for the attack (e.g., to probe the host and tune the exploit). Table 1 and Figure 1 summarize all of the variables as well as the states of these variables, which were used to create the scenarios. The overall hypotheses are that that those variables varied over the scenarios significantly influence the probability of success in remote arbitrary code execution attacks, and that this model is well-suited for predictions of success in remote arbitrary code execution attacks.

Table 1. Variables included in the attack scenarios.

Variable	**States studied**	**Description**
Proxy	Yes/No	If a filtering proxy, e.g. a filtering web proxy, is between the attacker server and the client.
DPI	Yes/No	If a deep-packet-inspection firewall is located between the attacker and the targeted server.
AccessControl	Yes/No	If the attacker can authenticate itself as a legitimate user of the service that is exploited in the attack. E.g., this variable is true if the attacked service is the SMB service (file and printer sharing) and the attacker is a part of the service's windows domain.
NX	Yes/No	If non-executable memory protection is activated on the targeted machine and used for the service attacked, e.g., DEP on a Windows machine or PaX on a Linux machine.
ASLR	Yes/No	If address space layout randomization is activated on the targeted machine.
Vulnerable	Yes	The targeted software has a high-severity vulnerability (as defined by CVSS (Mell et al., 2007)).
Connectivity	Yes	The attacker can send requests to the targeted service, e.g. because the firewall allows such connections.
CraftResponse	Yes	The attacker can craft (malicious) responses to the client, e.g., by luring the user of a web browser to a website controlled by the attacker.
Attacker	Resourceful	The attacker is a professional penetration tester with access to open and commercially available tools, and with one week to prepare the attack.

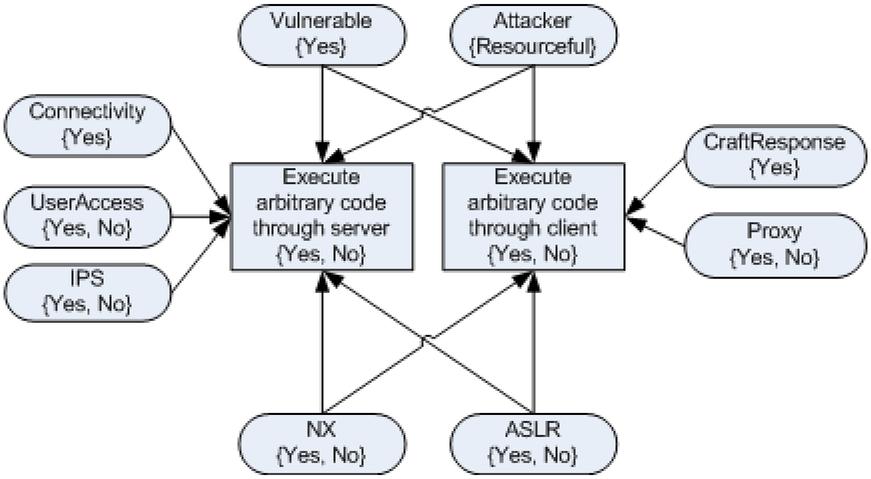


Figure 1. Variables included in the attack scenarios and dependencies investigated.

3 Synthesizing expert judgments

There is a substantial amount of research on how to combine, or synthesize, the judgment of multiple experts to increase the calibration of the estimates used. These techniques include the following: consensus methods (Fink et al., 1984; Ashton, 1985), the Cochran-Weiss-Shanteau index (Weiss and Shanteau, 2003), self-proclaimed expertise (Abdolmohammadi and Shanteau, 1992), experience (Shanteau et al., 2002), certifications (Shanteau et al., 2002), peer-recommendations (Shanteau et al., 2002), and Cooke's classical method (Cooke, 1991). There is little research that compares the accuracy that these methods yield. However, research has shown that groups of individuals assess an uncertain quantity better than the average expert, while the best individuals in the group are often better calibrated than the group as a whole (Clemen and Winkler, 1999). The scheme used to combine judgments in this research is the one used in the classical model of Cooke (Cooke, 1991). Cooke's model is a generic method for combining expert judgments that has been applied to a number of different domains. Experience from applications of Cooke's classical method has shown that it outperforms both the best expert and the "equal weight" combination of estimates. In an evaluation involving 45 studies, it performed significantly better than both alternatives in 27 studies and equally well as the best expert in 15 of the studies (Cooke, 2008).

In Cooke's classical method, *calibration* and *information* scores are calculated for the experts based on their answers to a set of seed questions, i.e., questions for which the true answer is known at the time of analysis. These two scores are used to define a *decision maker* that assigns weights to the experts based on their performance. These weights are used to create a single estimate on the variables of interest – in this case, the 24 attack scenarios. Cooke's classical method is briefly explained in Sections 3.1, 3.2 and 0. The reader is referred to (Cooke, 1991) for a detailed explanation of the method.

3.1 Calibration score

In the elicitation phase, the experts provide individual answers to the seed questions. The seed questions request that the respondents specify a probability distribution for a continuous variable for which the true value is uncertain to the respondent. This distribution is typically specified by stating its 5th, 50th, and 95th percentile values. This set of values yields four intervals over the percentiles *[0-5,5-50,50-95,95-100]* with probabilities of $p=[0.05,0.45,0.45,0.05]$. Because the seeds are realizations of these variables, a well-calibrated expert will have approximately 5% of the realizations in the first interval, 45% of the realizations in the second interval, 45% of the realizations in the third interval and 5% of the realizations in the fourth interval. If *s* is the distribution of the seed over the intervals, then the relative information of *s* with respect to *p* is the following: $I(s,p) = \sum_{i=1}^{4} \ln(s_i/p_i)$. This value indicates how surprised someone would be if one believed that the distribution was *p* and then learned that it was *s*.

If N is the number of samples/seeds, the statistic of 2*NI*(*s*, *p*) is asymptotically Chi-square distributed with three degrees of freedom. This asymptotic behavior is used to calculate the calibration (*Cal)* of expert *e* as the following: $Cal(e) = 1 - \chi_3^2(2N\, I(s,p))$. The calibration measures the statistical likelihood of a hypothesis. The hypothesis tested is that realizations of the seeds (*s*) are sampled independently from distributions that agree with the expert's assessments (*p*).

3.2 Information score

The second score used to weight experts is the information score, i.e., how precise and informative the expert's distributions are. This score is calculated as the deviation of the expert's distribution from some meaningful background measure. In this study, the background measure is a uniform distribution over [0,1].

If b_i is the background density for seed $i \in \{1,\ldots,N\}$ and $d_{e,i}$ is the density of expert *e* on seed *i*, the information score for expert *e* is calculated as the following: $\mathrm{inf}(e) = \frac{1}{N}\sum_{i=1}^{N} I(d_{e,i}, b_i)$, which is the relative information of the experts' distribution with respect to the background measure.

3.3 Constructing a decision maker

The classical method rewards experts who produce answers that have a high calibration (high statistical likelihood) and a high information value (low entropy). A strictly proper scoring rule is used to calculate the weights of the decision maker. If the calibration score of the expert *e* is at least as high as a threshold value, then the expert's weight is obtained as the following: $w(e) = Cal(e) * Inf(e)$. If the expert's calibration is less than the threshold value, the expert's weight is set to zero, a situation that is common in practical applications.

The threshold value corresponds to the significance level for the rejection of the hypothesis that the expert is well-calibrated. This value is the value that would optimize a virtual decision maker if it were added to the expert pool and had its weight calculated as one of the actual experts. When the threshold value is resolved, the normalized value of the expert weights *w(e)* is used to combine their estimates of the uncertain quantities of interest.

4 Method

4.1 Seed questions

Since the experts' performance in answering the seed questions is used to weight the experts, it is critical that the seeds are correct and are in the same domain as the variables that are studied. They need to be drawn from the relevant domain of expertise but do not need to be directly related to questions of the study (Cooke, 1991).

Naturally, the robustness of the weights that are given to individual experts depends on the number of seeds used. Experience shows that eleven seed questions are more than sufficient to see substantial differences in calibration (Cooke, 1991).

Two types of seed questions were used in this study. For the first type, questions (cf. #1-5 in Table 2) were drawn from the National Vulnerability Database (NVD) (NIST Computer Security Resource

Center, 2011) and concern statistics on known vulnerabilities in software products. The second type of question concerns the effectiveness of protective measures for buffer overflow vulnerabilities and was taken from (Wilander and Kamkar, 2003). Questions of the second type (cf. #5-11 in Table 2) asked the respondents to estimate how efficient protective measures were against 20 forms of attack that were described together with the questions.

Table 2. Seed questions and their realization values.

#	Question summary	Realization (%)
1	How many of the high-severity vulnerabilities published in 2010 have a full impact on Confidentiality, Integrity and Availability?	57
2	How many of the medium-severity vulnerabilities published in 2010 have a full impact on Confidentiality, Integrity and Availability?	6
3	How many of the vulnerabilities published in 2010 that can be exploited remotely require that the attacker bypass some authentication mechanism first?	9
4	How many of the vulnerabilities published in 2010 that can be exploited remotely and require that the attacker bypass some authentication mechanism first is of severity-rating high?	15
5	How many of the vulnerabilities published in 2010 that can be exploited remotely are of severity-rating high?	52
6	What is the probability that an attack (selected randomly from the 20 listed) will be prevented if Libverify and Libsafe are used?	0
7	What is the probability that an attack (selected randomly from the 20 listed) will be halted if Libverify and Libsafe are used?	20
8	What is the probability that an attack (selected randomly from the 20 listed) will be prevented if ProPolice is used?	40
9	What is the probability that an attack (selected randomly from the 20 listed) will be halted if ProPolice is used?	10
10	What is the probability that an attack (selected randomly from the 20 listed) will be prevented if Stackguard's terminator canary is used?	0
11	What is the probability that an attack (selected randomly from the 20 listed) will be halted if Stackguard's terminator canary is used?	15

4.2 The domain experts

Studies of expert calibrations have concluded that experts are well-calibrated in situations with learnability and with ecological validity (Bolger and Wright, 1994). Learnability is facilitated by the existence of models of the domain of interest; the possibility of expressing judgments in a coherent and quantifiable manner that can be verified; and the opportunity to learn from historic predictions and outcomes. Ecological validity is present if the expert is used to make judgments of the type that are requested.

In the context of this study, the above reasoning implies that good candidates are researchers and penetration testers in the security field. These individuals can be expected to be experienced in reasoning about the success or failure of attacks under different conditions and are expected to observe the outcomes of attempts. They also make judgments in their line of work (i.e., provide ecological validity).

To identify suitable respondents, articles published in the SCOPUS database (Elsevier B.V., 2011), INSPEC or Compendex (Elsevier Inc, 2011) between January 2005 and September 2010 were reviewed. Authors who had written articles in the information technology field with any of the words: “remote code execution”, “run arbitrary code”, “execute arbitrary code”, “arbitrary code execution”, “buffer overflow”, “buffer overrun” or “exploit code” in the title, abstract or keywords were identified. If their contact information could be found, they were added to the list of potential respondents, resulting in a sample of 964 individuals.

After the exclusion of individuals for which no contact information could be found and a manual review of their publications' topicality, a sample of 545 individuals was assessed. Contact information for approximately 110 of these individuals turned out to be incorrect or outdated, resulting in approximately 445 invitations reaching their destination.

A web survey was conducted during five weeks in December 2010 to January 2011. Out of approximately 445 researchers who were invited to take the survey, 119 opened it and 19 submitted answers to the survey's questions. A response rate of this magnitude is to be expected of an advanced survey such as this one. As recommended by (Cavusgil and Elvey-Kirk, 1998), motivators were presented to the respondents invited to the survey: i) helping the research community as whole, ii) the possibility to win a gift certificate for academic literature, and iii) being able to compare their answers to other experts after the survey was completed. One respondent provided contradictory and incomplete answers to the questions. After being unsuccessful in confirming these answers with this respondent, the respondent was excluded from further analysis, resulting in 18 usable surveys from researchers.

Additionally, practitioners were identified based on peer recommendations from notable practitioners in Sweden. Three practitioners, all with substantial experience in security exploits, participated in the study. Because practitioners are less likely to be as familiar with questionnaires in general and probability density functions in particular, these three respondents were given instructions on how to answer the survey during personal meetings in February and March 2011. Apart from the personal meetings, the participating practitioners answered the questionnaire in the same manner as the invited sample of researchers.

Thus, together with the three practitioners' surveys, the total number of respondents was 21.

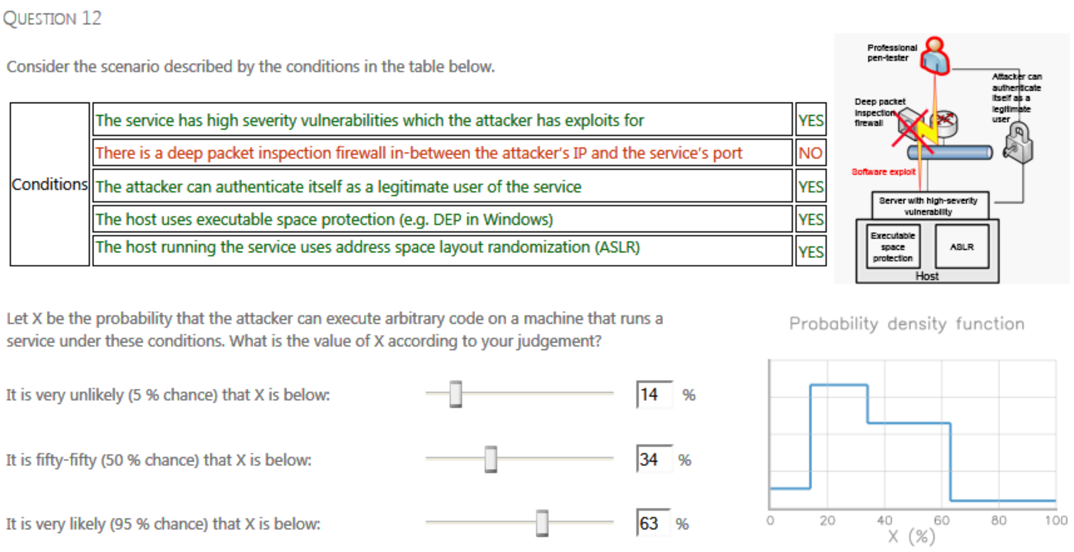
4.3 Elicitation instrument

The web survey comprised four parts, each beginning with a short introduction to the section. First, the respondents were given an introduction to the survey that explained the purpose of the survey and its outline. In this introduction, they confirmed that they were the person who had been invited and provided information about themselves, e.g., their number of years of experience in the field of research. Second, the respondents received training regarding the answering format used in the survey. After confirming that this format was understood, the respondents proceeded to its third part. Third, both the seed questions and the questions of the study were presented to the respondents. Finally, the respondents were asked to provide qualitative feedback on the survey and the variables that it covered.

Questions in section 3 of the survey were described through scenarios that detailed conditions for an attack. Summaries of the scenarios in the seed questions can be found in Table 1; conditions for the scenarios of interest in this study are described in section 2 of the paper.

For each scenario, the respondent was asked to provide a probability distribution that expressed the respondent's belief. As is customary in applications of Cooke's classical method (cf. Section 3), this probability distribution was specified by setting the 5th percentile, the 50th percentile (the median), and the 95th percentile for the probability distribution. In the survey, the respondents specified their distribution by adjusting sliders or entering values to draw a dynamically updated graph over their probability distributions. The three points specified by the respondents defined four intervals over the range [0, 100]. The use of graphical formats is known to improve the accuracy of elicitation (Garthwaite et al., 2005). Figures and colors were also used to complement the textual questions and to make the questions easier to understand. In Figure 2, the format presented to respondents is exemplified. A larger, generic figure that described the survey's variables could also be found at the top of each section, along with introductory text.

Elicitation of probability distributions is associated with a number of issues (Garthwaite et al., 2005). Efforts were therefore made to ensure that the measurement instrument was of sufficient quality. After careful construction, the survey was qualitatively reviewed during a personal session with an external respondent representative of the population. This session was divided into two parts. First, the respondent was given the task of filling in the survey, given the same amount of information as someone doing it remotely. After this task, discussions followed regarding the instrument quality. The qualitative review resulted in some minor improvements with respect to the phrasing of questions.



QUESTION 12

Consider the scenario described by the conditions in the table below.

Conditions	The service has high severity vulnerabilities which the attacker has exploits for	YES
	There is a deep packet inspection firewall in-between the attacker's IP and the service's port	NO
	The attacker can authenticate itself as a legitimate user of the service	YES
	The host uses executable space protection (e.g. DEP in Windows)	YES
	The host running the service uses address space layout randomization (ASLR)	YES

Let X be the probability that the attacker can execute arbitrary code on a machine that runs a service under these conditions. What is the value of X according to your judgement?

It is very unlikely (5 % chance) that X is below: 14 %

It is fifty-fifty (50 % chance) that X is below: 34 %

It is very likely (95 % chance) that X is below: 63 %

Professional pen-tester
Attacker can authenticate itself as a legitimate user
Deep packet inspection firewall
Software exploit
Server with high-severity vulnerability
Executable space protection
ASLR
Host
Probability density function
0
20
40
60
80
100
X (%)

Figure 2. Example of question and answering format in the survey

Before this qualitative review, the question format had been tested in a pilot study on other security parameters. In that pilot study, a randomized sample of 500 respondents was invited; 34 of these respondents completed the pilot during the week it was open. The questions in this pilot survey were presented in the same way as in the present survey. A reliability test using Cronbach's alpha (Cronbach and Shavelson, 2004; Cronbach, 1951) was performed using four different ways to phrase the questions for one variable. Results from this test showed a reliability value (alpha) of 0.817, which indicated good internal consistency of the instrument.

5 Results

5.1 Respondents' performance

As in many other studies that involve expert judgment, many of the experts were poorly calibrated on the seed questions. Their calibration scores varied between $3.211*10^{-14}$ and 0.6362, with a mean of 0.004255, and their information scores varied between 0.0658 and 1.847, with a mean of 0.7879.

Cooke's classical method aims to identify those respondents whose judgment is well calibrated and informative. The virtual decision maker was optimized at a threshold level (significance level) of 0.0007985. Four experts passed this threshold level and were assigned weights. They received the weights 0.8459, 0.1279, 0.02483, and 0.001361 after normalization. All four were researchers; their average experience from research on arbitrary code attacks was 12 years. As noted in Section 3.3, it is not uncommon that a substantial number of respondents receive a weight of zero with this method.

5.2 *Success rates of arbitrary code execution attacks*

The respondents' weights were used to construct the estimates of the virtual decision maker's estimates of success rates. In other words, the estimates described in this section represent the estimate of a virtual expert that is obtained by weighting the individual estimates of the respondents according to Cooke's method. The estimated distributions were assumed to be distributed in the same way that they were presented to the respondents, i.e., as depicted in the histograms over the four ranges that they constructed with their answers (c.f. Section 4.3). Note that certain variables are kept constant over the scenarios (c.f. Section 2).

5.2.1 *Server-side attacks*

As depicted Table 3, the synthesized estimates show clear differences among the scenarios. The median for the scenarios varies between 10 and 75 percent; the value at the 5th percentile varies between 1 and 17 percent, and the value at the 95th percentile varies between 48 and 94 percent. As one might expect, scenario 1 has the lowest median (10%) and expected (15%) success rate. Scenario 16 has, as one might expect, the highest success rate.

Table 3. Attack scenarios for server-side attacks.

Scenario	**Access Control**	**DPI**	**NX**	**ASLR**	**Low (5%)**	**Median (50%)**	**High (95%)**	**Expected (Mean)**
1	Yes	Yes	Yes	Yes	1	10	51	15
2	Yes	Yes	Yes	No	4	15	60	20
3	Yes	Yes	No	Yes	6	20	62	24
4	Yes	Yes	No	No	6	26	69	32
5	Yes	No	Yes	Yes	4	21	48	24
6	Yes	No	Yes	No	4	25	56	27
7	Yes	No	No	Yes	4	30	63	33
8	Yes	No	No	No	5	41	86	43
9	No	Yes	Yes	Yes	7	36	79	41
10	No	Yes	Yes	No	7	38	79	41
11	No	Yes	No	Yes	5	27	68	31
12	No	Yes	No	No	14	69	94	65
13	No	No	Yes	Yes	11	45	88	48
14	No	No	Yes	No	14	66	89	59
15	No	No	No	Yes	15	50	89	52
16	No	No	No	No	17	75	94	67

5.2.2 *Client-side attacks*

Table 4Table **4** lists the virtual decision maker's estimates for the eight attack scenarios considered for client-side attacks. In terms of the expected success rate, the difference between the most secure scenario (#17) and the least secure scenario (#24) is 24 percentiles. The low success rates associated with the server-side attacks where the attacker cannot gain user access is not present in these scenarios – the data received by the client are implicitly trusted by it.

Table 4. Attack scenarios for client-side attacks.

Scenario	Proxy	NX	ASLR	Low (5%)	Median (50%)	High (95%)	Expected (Mean)
17	Yes	Yes	Yes	7	38	84	43
18	Yes	Yes	No	10	43	89	47
19	Yes	No	Yes	12	48	94	52
20	Yes	No	No	15	53	94	55
21	No	Yes	Yes	4	54	95	56
22	No	Yes	No	15	58	94	59
23	No	No	Yes	18	63	95	62
24	No	No	No	20	72	95	67

5.3 Variables' influence on the success rate of exploits

This study varies four variables in each set of scenarios. The variation over the scenarios supports the hypothesis that these variables are relevant for the success rate. Table 5 shows their mean influence on the estimates. These values are the mean difference obtained when comparing scenarios in which the variable is in the state of "true" with those scenarios in which the variable is in the state "false" and all other variables remain in the same state. For example, the values for *AccessControl* in the server-side scenarios are obtained as the mean value of the difference between the following scenarios: 1 and 9; 2 and 10; 3 and 11; and so on. A combination of variables (e.g., "DPI & NX") shows the mean influence that the combination has when compared to the individual influences that they have alone. A positive value for a combination indicates that the measures cancel each other out to an extent; a negative value indicates that the combined measures complement each other and that the joint effect is greater than the sum of the individual measures.

Table 5. Mean influence of the variables on the success rate (in percent).

Scenarios	Variable	Low (5%)	Median (50%)	High (95%)	Expected (Mean)
Server	AccessControl	-7.00	-27.25	-23.13	-23.25
	DPI	-3.00	-14.00	-6.38	-10.50
	NX	-2.50	-10.25	-9.38	-9.00
	ASLR	-2.25	-14.50	-9.88	-10.75
	AccessControl & DPI	+3.00	+2.50	+3.63	+1.50
	AccessControl & NX	+0.50	-1.25	-6.88	-2.50
	AccessControl & ASLR	+1.25	+8.00	-1.88	+4.25
	DPI & NX	-0.50	-0.50	+3.38	+0.25
	DPI & ASLR	-0.75	+0.75	-0.63	-1.00
	AccessControl & DPI & NX	-1.00	+1.50	+2.88	+0.75
	AccessControl & DPI & ASLR	+0.25	+0.25	+4.38	+1.00
	DPI & NX & ASLR	+0.75	+3.75	+0.63	+3.25
	AccessControl & I PS & NX & ASLR	-1.75	-5.25	-4.88	-4.25
Client	Proxy	-3.25	-16.25	-4.50	-11.75
	NX	-7.25	-10.75	-4.00	-7.75
	ASLR	-4.75	-5.75	-1.00	-3.75
	Proxy & NX	+2.25	+0.75	-3.50	-0.75
	Proxy & ASLR	+1.75	+0.75	-1.50	+0.25
	NX & ASLR	-2.25	+1.25	-1.0	+0.25
	Proxy & NX & ASLR	+2.25	-1.25	-1.5	-0.75

As can be seen from Table 5, restriction of access influences server-side attacks the most wheras the presence of a filtering proxy shows the most influence on client-side attacks. The respondents seem to perceive the studied variables as fairly independent, i.e., the effects from combinations of them are small.

6 Discussion

6.1 The expert judgment analysis

Eleven seed questions were used to evaluate the calibration and information scores. These seed questions are of two types. The first type of seed question is drawn from a vulnerability database and concerns the characteristics of known vulnerabilities. The second type is drawn from an empirical peer-reviewed study (Wilander and Kamkar, 2003) on the types of exploits that different countermeasures protect against. Both of these types of questions are strongly related to the expertise that is required to answer the question of interest. A concern about the survey's validity could be that these sources are available to the respondents, who could have used them to identify the answers to the seed questions. However, no indications of this concern were seen in the answers received or in the feedback from the respondents.

The calibration scores show that many experts in the field are poorly calibrated, i.e., their estimates do not match empirical observations well. This observation suggests that sorting out well-calibrated experts is worthwhile. Four respondents were assigned weights when the virtual decision-maker was optimized. When using this method to assign weights, it is appropriate to perform a robustness test on the solution (Cooke, 1991). These tests are performed with respect to both seed variables and experts by removing one at a time and by investigating the impact of the omission (Cooke, 1991). Such tests were performed and no undue influence was identified.

6.2 Validity and reliability of the elicitation instrument

Cooke (1991) suggests that seven guidelines should be used when data are elicited from experts: i) formulate clear questions, ii) use an attractive format for the questions and a graphical format for the answers, iii) perform a dry run, iv) have an analyst present during the elicitation, v) prepare an explanation of the elicitation format and how answers will be processed, vi) avoid coaching and vii) keep elicitation sessions to less than one hour long.

This study follows all of these guidelines except for iv), which is to have an analyst present during the elicitation. The invited researchers were given contact information to the research group when invited to the survey, which they were encouraged to use if any questions arose. Practitioners were also introduced to the survey format personally. However, it is possible that the physical absence of the analysts suppressed some potential issues from being brought up during the elicitation. In the survey, the respondents were asked to comment on the clarity of the questions and the question format used. Based on the comments received, it appears as though the questions and the assumptions were fully understood.

6.3 Variables of importance to the success rate

The models used to describe attack scenarios in this study contained four variables for server-side attacks and three variables for client-side attacks. All these variables have an influence on the success rate. The result shows that the most influential countermeasures against server-side attacks are to make certain that attackers do not obtain access credentials to the service. If the attacker does not have access rights for the service, the expected success rate is decreased by 23 percentiles on average. However, restricting access can be difficult, for example, in the case of public services. Address space layout randomization, non-executable memory, and deep-packet inspection also lower the attack success rate significantly. Taken together, these three countermeasures lower the expected success rate by 26-28 percentiles. For client-side attacks, a filtering proxy is the most effective; address space layout randomization and space execution prevention is less potent than on server-side attacks.

The scenarios estimated in this study did not specify all of the variables that could be relevant. The undefined variables (e.g., the type of service that is vulnerable) certainly vary among and within enterprises. As a result, it is impossible to say how much of the uncertainty arises from variations among unspecified variables in enterprises (i.e., aleatory uncertainty) and how much arises from the expert's lack of knowledge about arbitrary code attacks (i.e., epistemic uncertainty). However, it is reasonable to expect that both types of uncertainty contribute to the spread of the estimated intervals.

The variables included in this study were drawn from the literature with the assistance of domain experts with practical experience from arbitrary code execution attacks and the effectiveness of several of those variables was evaluated in a quantitative pilot study (Holm et al, 2011). The hypothesis was that these variables make up a good model for predicting the probability of successful remote arbitrary code execution. The respondents of the survey were asked to improve this model by replacing one of the variables with a new variable of their own choice. Three of the respondents suggested changes to the model. In terms of the calibration score, these three variables are ranked third, eighth and eighteenth. Two of those respondents (ranked third and eighth) suggested that the implementation of NX should be detailed in the model, e.g., if it is the implementation for Linux Red Hat 4.1 or Windows XP SP2. One respondent (ranked eighteenth) would like to replace ASLR with the existence of a host-based intrusion detection system in the targeted machine. The fact that only three of the 21 respondents suggested changes to the model indicates that it successfully captured the most important variables. However, future work in this field could add more detail to the scenario descriptions to identify the differences between different NX implementations and to investigate the impact of host-based intrusion detection.

7 Conclusions

The synthesized judgment of domain experts is that the most effective measure against server-side arbitrary code execution attacks is to implement access controls that limit the functionality that attackers can use. However, deep-packet inspection firewalls and measures available in operating systems (*ASLR* and *NX*) also lower the probability of successful compromise. For client-side attacks, where an application client is exposed to malicious data, the most effective countermeasure is the use of a filtering proxy. Operating system measures do not have as strong effects on attacks against clients. Decision-makers in enterprises should consider these effects when they contemplate measures against code injection attacks.

However, while these synthesized judgments provide valuable input to decision-makers and researchers, they come with a substantial amount of uncertainty. Further research could add more detailed variables to the attack scenarios to remove aleatory uncertainty. Also, this would enable more detailed data collection from experiments or observations to remove epistemic uncertainty. The results from this study can provide valuable information to future studies in this direction e.g., the approximate importance of the studied variables and that they are perceived to be fairly independent.

8 References

Abdolmohammadi, M.J. et al. 1992. Personal attributes of expert auditors. Organizational Behavior and Human Decision Processes, 53(2), p.158–172.

Ashton, A.H. 1985. Does consensus imply accuracy in accounting studies of decision making? The Accounting Review, 60(2), p.173–185.

Bolger, F., & Wright, G. (1994). Assessing the quality of expert judgment: Issues and analysis. Decision Support Systems, 11(1), 1-24.

Cavusgil, S. T., & Elvey-Kirk, L. A. (1998). Mail survey response behavior: A conceptualization of motivating factors and an empirical study. European Journal of Marketing, 32(11/12), 1165–1192. MCB UP Ltd.

Clemen, R. T., & Winkler, R. L. (1999). Combining probability distributions from experts in risk analysis. Risk Analysis, 19(187), 187-204.

Cooke, R. (2008). TU Delft expert judgment data base. Reliability Engineering & System Safety, 93(5), 657-674.

Cooke, Rm. (1991). Experts in uncertainty: opinion and subjective probability in science, Oxford University Press, Oxford.

Cooke, Rm, & Goossens, L. (2004). Expert judgement elicitation for risk assessments of critical infrastructures. Journal of Risk Research 7(6), 643-656.

Cowan, C., Wagle, P., Pu, C., Beattie, S., & Walpole, J. (2003). Buffer Overflows : Attacks and Defenses for the Vulnerability of the Decade. Foundations of Intrusion Tolerant Systems, 2003 Organically Assured and Survivable Information Systems, pp. 227-237.

Cronbach, L.J. (1951). Coefficient alpha and the internal structure of tests. Psychometrika, 16(3), 297–334. Springer.

Cronbach, Lee J., & Shavelson, R. J. (2004). My Current Thoughts on Coefficient Alpha and Successor Procedures. Educational and Psychological Measurement, 64(3), 391-418.

Elsevier B.V. (2011). Scopus. Retrieved from http://www.scopus.com/.

Fink, a et al., 1984. Consensus methods: characteristics and guidelines for use. American journal of public health, 74(9), pp.979-83.

Garthwaite, P. H., Kadane, J. B., & O'Hagan, A. (2005). Statistical methods for eliciting probability distributions. Journal of the American Statistical Association, 100(470), 680-701.

Holm, H. et al., 2011. Expert assessment on the probability of successful remote code execution attacks. In *Proceedings of 8th International Workshop on Security in Information Systems - WOSIS 2011*. Beijing.

Homer, J., Manhattan, K., Ou, X., & Schmidt, D. (2010). A Sound and Practical Approach to Quantifying Security Risk in Enterprise Networks. Technical report, Kansas State University, Computing and Information Sciences Department. August 2009.

Mell, P., Scarfone, K., & Romanosky, S. (2007). A Complete Guide to the Common Vulnerability Scoring System Version 2.0. System, 1-23.

NIST Computer Security Resource Center (CSRC), U. S. D. of C. (2011). National Vulnerability Database. Retrieved April 28, 2011, from www.nvd.nist.org.

Patsos, D., Mitropoulos, S., & Douligeris, C. (2010). Expanding topological vulnerability analysis to intrusion detection through the incident response intelligence system. Information Management & Computer Security, 18(4), 291-309..

Ryan, J. J. C. H., Mazzuchi, T. a, Ryan, D. J., Lopez de la Cruz, J., & Cooke, Roger. (2010). Quantifying information security risks using expert judgment elicitation. Computers & Operations Research, 1-11. Elsevier.

Sawilla, R., & Ou, Xinming. (2008). Identifying critical attack assets in dependency attack graphs. 13th European Symposium on Research in Computer Security (ESORICS) (pp. 18-34). Springer.

Scarfone, K., & Mell, P. (2007). Guide to intrusion detection and prevention systems. Nist Special Publications, 800(94).

Shacham, H., Page, M., Pfaff, B., & Goh, E. (2004). On the effectiveness of address-space randomization. ACM conference on Computer and communications security, 298. New York, New York, USA: ACM Press.

Weiss, D.J. & Shanteau, J., 2003. Empirical assessment of expertise. Human Factors: The Journal of the Human Factors and Ergonomics Society, 45(1), p.104.

Sommestad, T., Ekstedt, M., & Johnson, P. (2010). A Probabilistic Relational Model for Security Risk Analysis. Computers & Security 29(6), 659-679.

Wilander, J., & Kamkar, M. (2003). A comparison of publicly available tools for dynamic buffer overflow prevention. Proceedings of the 10th Network and Distributed System Security Symposium (p. 149–162).

Younan, Y. (2008). Efficient countermeasures for software vulnerabilities due to memory management errors. PhD thesis, Katholieke Universiteit Leuven.