Security Analysis of VPN Configurations in Industrial Control Environments

Sanaz Rahimi, Mehdi Zargham

HAL Id: hal-01571782

https://inria.hal.science/hal-01571782

Submitted on 3 Aug 2017

Chapter 6

SECURITY ANALYSIS OF VPN CONFIGURATIONS IN INDUSTRIAL CONTROL ENVIRONMENTS

Sanaz Rahimi and Mehdi Zargham

Abstract   Virtual private networks (VPNs) are widely recommended to protect otherwise insecure industrial control protocols. VPNs provide confidentiality, integrity and availability, and are often considered to be secure. However, implementation vulnerabilities and protocol flaws expose VPN weaknesses in many deployments. This paper uses a probabilistic model to evaluate and quantify the security of VPN configurations. Simulations of the VPN model are conducted to investigate the trade-offs and parameter dependence in various VPN configurations. The experimental results provide recommendations for securing VPN deployments in industrial control environments.

Keywords: Control systems, virtual private networks, security analysis

1.   Introduction

Virtual private networks (VPNs) are widely used to provide secure communications over insecure public networks. VPNs provide security services such as confidentiality, integrity and availability by creating encrypted tunnels between the communicating parties.

VPNs are recommended in the literature and by many critical infrastructure protection standards to secure process control, SCADA and automation protocol communications [14–16, 21, 22]. Although these protocols are generally very reliable, they were not designed to resist malicious attacks. As a result, it is recommended to wrap industrial protocols such as DNP3 [18], 61850 [13] and Modbus [19] within VPN tunnels to protect them from unauthorized access. These configurations supposedly offer confidentiality, integrity and availability [22], but little work has focused on the secure configuration of VPN tunnels and the maintenance required for their secure operation [22].

VPNs are attractive targets for attackers. Because VPNs carry sensitive information over public networks, successfully breaking into a VPN tunnel enables an attacker to alter sensitive data and commands without physical access to the industrial facility. If other protection mechanisms such as strong access control are not deployed properly, the attacker can gain access to the internal SCADA systems through a VPN tunnel. Also, as industrial systems implement more security mechanisms, VPNs can become the weakest link in the security chain.

VPNs have several vulnerabilities. According to Hills [12], most VPN implementations suffer from serious security flaws that can be easily exploited by attackers to fabricate, intercept, modify and interrupt traffic. Some of these vulnerabilities are implementation specific; they are the result of flaws in a specific protocol implementation due to bad coding, incomplete implementation or poor implementation choices for conditions that are not specified in the standard. However, vulnerabilities in the underlying protocols cannot be addressed by good implementation [12]. Finally, as recent incidents have shown, sophisticated malware [6] can stealthily modify the configurations of a control system (including a VPN) and seriously impact its operation. The solutions to these security problems are proper configuration, continual configuration validation and regular maintenance, all of which are effective only if system administrators fully understand the internal details of the protocols.

This paper models VPNs using stochastic activity networks [25], which are an extended form of Petri nets [3], and analyzes the probability of a successful breach against various VPN configurations. VPN security is quantified for different choices of parameters such as key length, mode of operation, number of users and maintenance frequency. The results provide recommendations for securely deploying VPNs in industrial control environments.

2.   VPNs and VPN Vulnerabilities

VPNs are categorized with respect to their layer, i.e., transport layer (SSL), network layer (IPSec) and link layer (L2TP). This paper focuses on IPSec VPNs. As an underlying VPN protocol, IPSec [23] provides confidentiality and integrity services in the network layer (i.e., on a per packet basis) using two main sub-protocols (AH and ESP) in two different modes (transport and tunnel). The detailed description of IPSec is beyond the scope of this paper. We only describe the features that are relevant to our discussion. Interested readers are referred to [23] for additional information about IPSec.

2.1   IPSec

IPSec provides security services via the Authentication Header (AH) and Encapsulation Security Payload (ESP) protocols. AH provides integrity by adding the HMAC [5] of the entire packet (payload and full IP header); however, it does not provide confidentiality because it leaves the packet in plaintext. ESP encrypts the packet payload and some fields of the IP header, and adds the

ESP header and trailer to the IP packet, providing confidentiality and limited integrity.

IPSec can operate in the transport or tunnel modes. The transport mode is used when end-to-end security is desired, and both end nodes support IPSec. In the transport mode, the original IP header is preserved for routing purposes and the AH/ESP headers are added under the IP header. The tunnel mode is used when the end machines do not support IPSec or when the identities of the communicating entities have to stay hidden. In the tunnel mode, the entire IP packet is encrypted and a new IP header is added to the packet. The gateway on the border of each organization provides the security services by adding and removing IPSec headers.

Security Association (SA) is the concept used in IPSec for connection management between two communicating entities. An SA comprises a secure communication channel and its parameters, including the encryption algorithms, keys and lifetimes. Each SA is unidirectional and can provide one security service (AH or ESP). Two SAs are required for bidirectional communications.

IPSec uses the Internet Key Exchange (IKE) protocol to manage and exchange encryption keys and algorithms. IKE is a hybrid of three sub-protocols: Internet Security Association and Key Management Protocol (ISAKMP), Versatile Secure Key Exchange Mechanism for Internet (SKEME) and Oakley. ISAKMP provides the framework for authentication and SA management, but does not define the specific algorithms and keys. IKE uses the SKEME and Oakley protocols for key exchange and agreement with the acceptable cryptographic algorithms.

Because IKE is commonly used to establish VPN channels, many VPN vulnerabilities are in one way or another related to it. For a better understanding of these vulnerabilities, we provide an overview of IKE and its modes of operation.

IKE has three important modes: main mode, aggressive mode and quick mode. The main mode is used for authentication and key exchange in the initial phase of IKE. This phase assumes that no SA is present and that the two parties wish to establish SAs for the first time. It involves three pairs of messages. The first pair negotiates the security policy and encryption algorithms to be used. The second pair establishes the keys using the Diffie-Hellman key exchange protocol. The third pair authenticates peers using signatures or certificates. Note that the identities of the peers in the main mode are often their IP addresses.

The aggressive mode is also used for the initial key exchange, but it is faster and more compact than the main mode. This mode involves a total of three messages that contain the main mode parameters, but in a more compact form. Key and policy exchange are performed by the first two messages while the third message authenticates the initiator to the responder. Note that the identity of the responder (sent in the second message) is not protected, which is a vulnerability in the aggressive mode of operation.

The quick mode is used for the negotiations in the secondary phase of IKE. This mode assumes that the peers have already established SAs and that the exchange can update the parameters or renew the keys. The quick mode messages are analogous to those in the aggressive mode, but the payloads are encrypted. If the quick mode operates with the perfect-forward-secrecy option, the shared secrets are renewed with a fresh Diffie-Hellman exchange.

IKE authenticates peers using a pre-shared key (PSK), public key encryption or digital signature. In the PSK method, which corresponds to the traditional username/password authentication, the peers share a secret through a back channel and exchange the hash value of the secret for authentication. Unfortunately, although this method has known vulnerabilities, it is the only mandatory authentication method according to the RFC [11]. Public key encryption is another method of authentication in which the peers digitally sign their identities; however, the keys are required to be provided beforehand by some other means. Digital certificates may also be used for authentication in IKE; in this mode, the peers exchange certificates to mutually authenticate each other.

2.2   VPN Vulnerabilities

VPNs have several vulnerabilities. The common username enumeration vulnerability refers to an implementation flaw in which the username/password authentication mechanism responds differently to invalid usernames and passwords. By exploiting this vulnerability, an attacker can identify valid usernames that can be used later in password discovery.

When IKE is used in the aggressive mode with a pre-shared key (PSK), the client sends an IKE packet to the server containing, among other things, the identity (username) of the client. The server then responds with another packet that contains the hash of the identity and the PSK (password). When an incorrect username is received, many VPN implementations send an error message, send a packet with NULL PSK or do not respond at all. This enables an attacker to infer whether or not a username exists by sending a single packet (enumerate username). Upon discovering the first username, attacker can generate likely usernames with the same pattern (e.g., first letter of the first name concatenated with the last name). When a VPN is used in the main mode, the identity is an IP address, not a username.

Hills [12] proposes that a secure VPN implementation return the hash of a randomly-generated password each time it receives an invalid username. However, this does not solve the problem because an attacker can still send two different packets with the same username; if two different hashes are received, then the attacker knows that the username does not exist and vice versa. Furthermore, attacker can delay these two packets using a number of packets for other usernames to flush a buffer that the server may employ to track such an attack. The solution is for the server to encrypt the username with a secret key (generated and kept on the server only for this purpose) and to return the hash

of this value. Thus, the server always responds to a username with a unique hash value, which foils the attack.

When the attacker discovers a valid username, he/she can receive the hash of the username's password from the server (using PSK in the aggressive mode). The attacker can then apply an offline algorithm to crack the hash value and obtain the password. Offline cracking can be very fast because the probabilistic model of VPN password hashing is not hidden. This poses a serious threat to short passwords. The vulnerability exists even if IKE operates in the main mode with PSK; it can be exploited by a man-in-the-middle attack (e.g., using DNS spoofing [20]) to gain the Diffie-Hellman shared secrets. The only difference is that, in the main mode, the identity of each peer is its IP address.

When a username/password pair is successfully found, the first phase of IKE is breached. If the VPN configuration does not require extra authentication, the breach is enough to setup a VPN channel with the server. In some other cases, the configuration requires an extra XAUTH step to complete the second phase of IKE, but this phase is vulnerable to a man-in-the-middle attack as mentioned in the standard [24]. The reason for the vulnerability is that XAUTH must be used in conjunction with the first phase of IKE; if this phase is not performed properly, XAUTH cannot provide any security guarantees. Therefore, an attacker who performs a man-in-the-middle attack would be able to authenticate successfully.

Other VPN implementation vulnerabilities include VPN fingerprinting (inferring information about a device from its behavior); insecure password storage (e.g., in the registry, as plaintext in memory, etc.); lack of account lockout; poor default configurations; and unauthorized modification of configurations. We do not consider these vulnerabilities in this paper because they are implementation specific and may require other flaws to be exploited in a successful attack (e.g., insecure password storage would be exploited by the installation of malware).

3.   VPN Security Model

This section describes the probabilistic model used to analyze VPN security. The model helps quantify the security of a protocol and provides suggestions for its secure implementation, configuration and operation. This is important because, as Hills [12] notes, VPN tunnels are misconfigured approximately 90% of the time.

VPN security is modeled using a stochastic activity network (SAN) [25], which is an extension of a Petri net [3]. The Mobïus tool [7] is used to specify the model and to perform numerical simulations. This section explains the details of the SAN model and its parameters; all times in the model are expressed in minutes.

The model comprises two sub-models (atomic models), one models the implementation and configuration of a VPN tunnel, and the other models its environment and operational details. The two sub-models are joined into a composed VPN model using the Rep/Join representation [7].
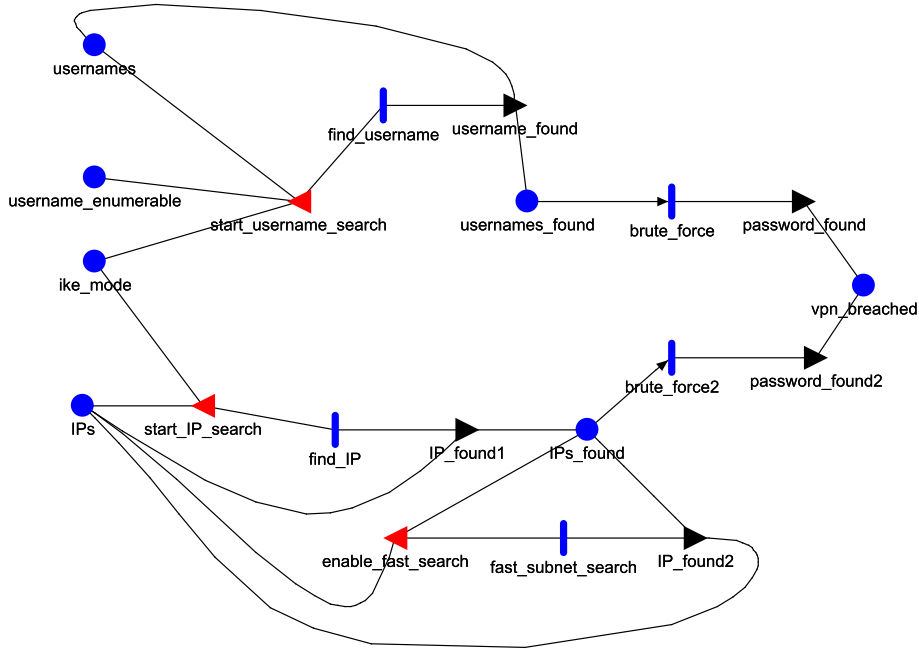
Figure 1.   Probabilistic model of a VPN.

The first atomic model (*ike*) models the weaknesses of the protocol (Figure 1). A global variable identifies if the VPN is operating in the main mode or in the aggressive mode. If the VPN is configured in the aggressive mode, an activity models the username enumeration attack. We consider usernames that consist of alphabetic characters and have a length of at most six characters (approximately 309 million possibilities).

If the roundtrip time between the scanner and server is in the order of tens of milliseconds [17] and a window of ten packets is used, then, on the average, it takes 1 ms to check for each username; so we assume that 1,000 usernames can be checked per second. If the roundtrip time is larger, an appropriate window can be chosen to achieve this rate. With this rate, it is possible to exhaustively scan the username space in approximately 3.5 days. A sophisticated attacker can do better using a fast connection to the server and/or an intelligent guessing algorithm after a username is found. However, in this paper, we consider an unsophisticated attacker in order to obtain an upper bound on VPN security. Note that, because username scanning does not typically cause account lockout, this process is not stopped by the server.

The rate of username enumeration is proportional to the number of system users (more users result in a faster enumeration using exhaustive search). This is modeled by multiplying the base rate (1 per 3.5 days = 1.94E-04 per minute) by the marking of (the number of tokens in) the *usernames* place. Whenever a username is found, it is moved from the pool of unknown user-

Table 1. Password complexity and password space size.

Password Complexity	Space Size
6 characters a-z	3.1E+08
6 characters a-z, A-Z, 0-9	5.7E+10
8 characters a-z	2.1E+11
8 characters a-z, A-Z, 0-9	2.1E+14

names to *usernames_found* using the output gate *username_found*. Note that *usernames_found* is a place that holds the number of usernames enumerated at a given point in time, whereas *username_found* is the output gate that transfers the discovered username from the unknown usernames place (*usernames*) to the *usernames_found* place.

When a username is found, the attacker can start an offline attack to obtain the password. To crack the password, the attacker has to hash different values exhaustively. The cracking speed for MD5 hashes using an AMD Athlon XP 2800+ is around 315,000 attempts per second ($\sim$1.9E+07 attempts per min) [12]. Since the cracking speed depends heavily on password complexity, the model is run using different password space sizes. Table 1 shows the size of the password space for different types of passwords.

The rate of successful attempts is proportional to the number of usernames enumerated, so the rate of the *brute_force* activity is multiplied by the marking of the place *usernames_found*. If a username/password pair is found, the VPN is breached. Other transactions (e.g., to setup a VPN tunnel after the breach) require negligible time compared with brute force or username enumeration. As result, after a username/password pair is found, a token is placed in *vpn_breached*.

The other possible mode of operation is the main mode. As mentioned before, in the main mode, the identities are the IP addresses of the peers. The space of 32-bit IP addresses is approximately fourteen times larger than the space of six-character usernames; thus, the *find_IP* activity that randomly searches the IP address space has a base rate that is fourteen times slower than *find_username*. However, upon finding a valid IP address, the attacker can perform subnet-based search, which makes it much faster to find other IP addresses (assuming, of course, that most of the clients are in the same subnet). Note that for the main mode to be enabled, at least one IP address must be found by random search. Upon finding a valid IP address, the attacker exhaustively searches the space of PSKs similar to the aggressive mode, placing a token in *vpn_breached* whenever an IP/PSK pair is found.

The second atomic SAN model captures the behavior of the VPN environment and its operational maintenance (Figure 2). VPNs are vulnerable to malware attacks [8]. In particular, malware can modify the configuration of a VPN tunnel in a stealthy manner. Since the VPN tunnel remains operational
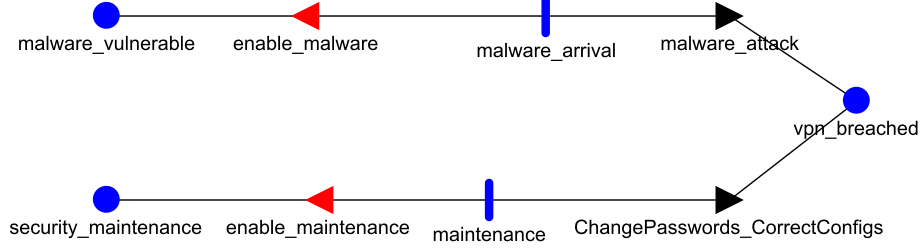
Figure 2. VPN malware infection and maintenance models.

after the modification, it is difficult for the system administrator to detect such an attack.

We model two environments, one in which malware attacks exist and the other where they do not. Malware can maliciously modify a VPN to send packets in plaintext. Therefore, the installation of malware is synonymous to a VPN breach.

The malware infection rate is hard to quantify in industrial systems. For the sake of argument, however, we choose an infection rate of once a month when malware is present and later show that this rate does not have a significant impact on VPN security. The activity *malware_arrival* models malware infections. Although malware can also retrieve unsecured passwords, we do not consider it to be a part of model because it is implementation specific.

VPN maintenance by the system administrator is a preventive and/or corrective action that can improve security. Maintenance involves changing passwords and checking for bad configurations. If a VPN configuration is modified by malware, a maintenance operation can secure the VPN by correcting the configuration and installing a patch that deals with the malware. Also, regularly changing passwords can mitigate exhaustive search attacks, helping secure the VPN. On the other hand, password changes do not affect username/IP enumeration, so this activity does not flush the *usernames_found* and *IPs_found* places in the model.

4. Experimental Results

This section presents the results obtained from simulation experiments using the SAN model. The primary goal was to investigate the probability that the VPN is not in the breached state. In SAN terminology, the reward variable (*security_probability*) is defined as the probability that the marking of the place *VPN_breached* is zero. The value of this probability for each configuration was studied for different time periods: one hour, three hours, twelve hours, one day, three days, one week, one month, three months and one year.

VPN security was also studied for different IKE modes (aggressive vs. main mode); password complexity (Table 1); numbers of users/machines (1, 10, 100 and 250); environments (with or without malware); and maintenance rates (once every week, month, three months, one year and no maintenance). Given
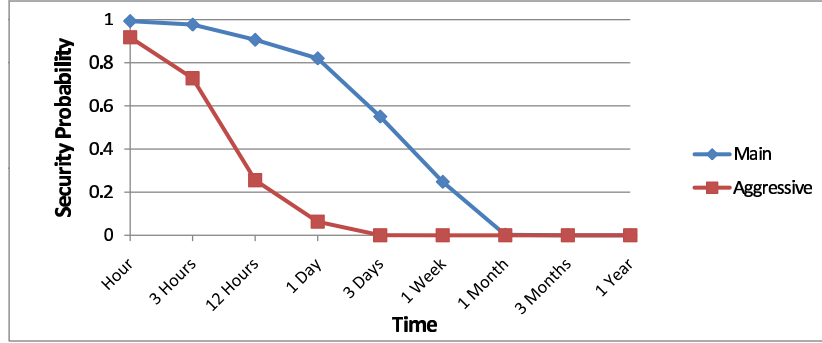
Figure 3.   VPN security in the aggressive mode versus the main mode.

the number of factors of interest, a large number of experiments are possible, but not all of them are meaningful. Therefore, we only performed experiments in which a few parameters of interest were changed at a time.

The first experiment compared VPN security in the aggressive mode versus the main mode. The main mode is generally more secure because, in order to perform offline password cracking, the attacker has to conduct a man-in-the-middle attack. Even if this attack is successful, the space of 32-bit IP addresses is larger than the space of usernames. The experiment assumed that the attacker can perform a man-in-the-middle attack; otherwise, the main mode is not vulnerable to the attack (i.e., the security probability is one at all times.)

The passwords (or PSKs) for both modes were selected from the space of six alphabetic characters (3.1E+08); the system was assumed to have ten users (usernames or IP addresses); security maintenance was not performed; and malware was not present. The results are shown in Figure 3. The security of a VPN tunnel diminishes over time. However, the security declines faster for the aggressive mode than for the main mode. The aggressive mode is less than 50% secure after six hours whereas the main mode reaches this level after about four days. Note also the short lifetime of a six-character alphabetic password for a VPN tunnel.

The second experiment studied the effect of password complexity on the overall security of a VPN (aggressive mode). To observe the effect of password complexity alone, maintenance and malware were switched off in this experiment. The system had ten different users.

The results are shown in Figure 4. As expected, the overall security of a VPN increases with password complexity. Note that eight-character alphanumeric passwords are secure for a much longer period of time, but even this type of password is less than 65% secure after one year. On the other hand, six-character alphanumeric passwords are less than 60% secure after just one day.

The third experiment studied the effect of maintenance frequency on VPN security. This experiment assumed that IKE was used in the aggressive mode, that the system had ten users and that six-character alphanumeric passwords
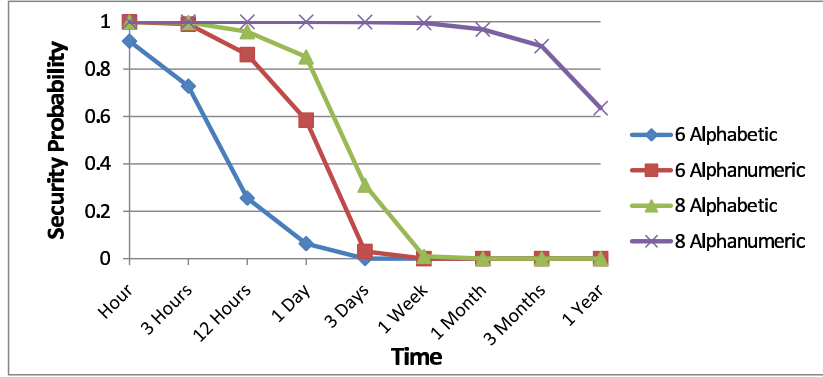
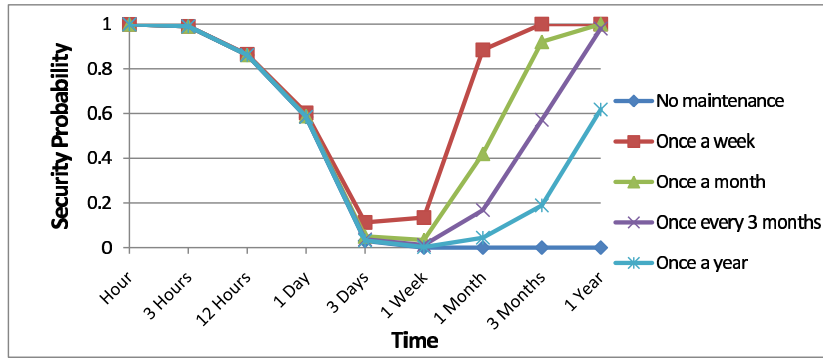*Figure 4.* Impact of password complexity on VPN security.

Figure 5. Impact of maintenance frequency on VPN security.

were used. The results in Figure 5 show that frequent maintenance can mitigate the effect of weak configurations. Note that the security probability reduces until it reaches a minimum before any maintenance starts; after this, the probability increases with frequent maintenance. Since the rate of maintenance is higher than the rate of password cracking in each case, the security probability reaches one at steady state. This does not mean that it is impossible to break the VPN tunnel as time passes; rather, it implies that the portion of time that the VPN tunnel is breached diminishes over longer time periods. Moreover, the declining security trend during the first few days can be repeated after each successful maintenance. The effect is not shown in Figure 5 because the security probability represents the steady state measure of security at any time.

The fourth experiment studied the effect of malware attacks versus weak VPN passwords. Two values of password complexity (six- and eight-character alphanumeric passwords) are plotted in Figure 6 with and without frequent malware attacks (once a month.) In the experiment, IKE used the aggressive mode and no maintenance was performed. A counterintuitive result from this
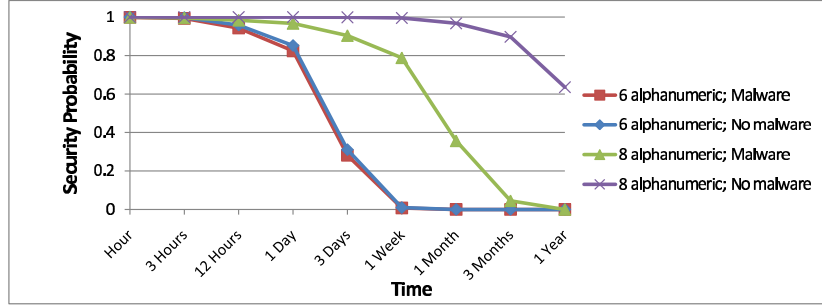
Figure 6.   Impact of malware on VPN security.

experiment is that malware infections have little impact on the security of a weakly-configured VPN because the dominant effect in this mode is the ability of an attacker to crack a six-character alphanumeric password. On the other hand, in the case of a strong password, frequent malware infections considerably weaken VPN security. Therefore, we conclude that the impact of a malware infection depends on the configuration of the VPN. If the rate of password cracking is higher than the rate of infection, malware has little impact on the system. As a result, priority must be given to secure a VPN configuration. Note that this study only considered the effect of malware on the security of the VPN tunnel. Malware infections have other negative security impacts that were not modeled in the experiment.
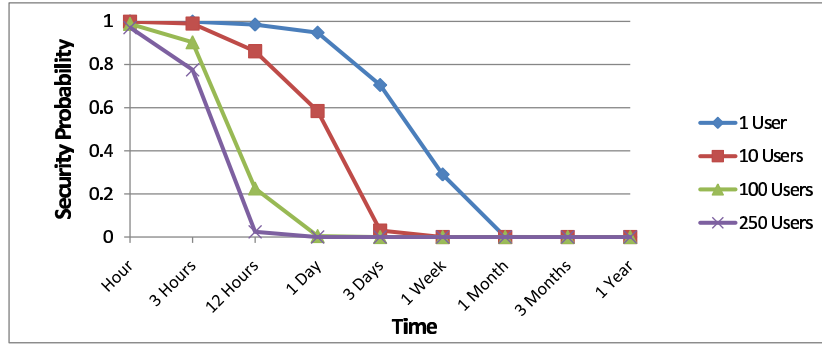

Figure 7.   Impact of the number of users on VPN security.

Next, we studied the effect of the number of users on overall VPN security. As shown in Figure 7, systems with large user populations are much less secure than systems with few users because an attacker has a higher chance of finding valid usernames/passwords (or IPs/PSKs.) This experiment assumed that IKE was used in the aggressive mode, that six-character alphanumeric passwords were used, security maintenance was not performed and malware was not present.
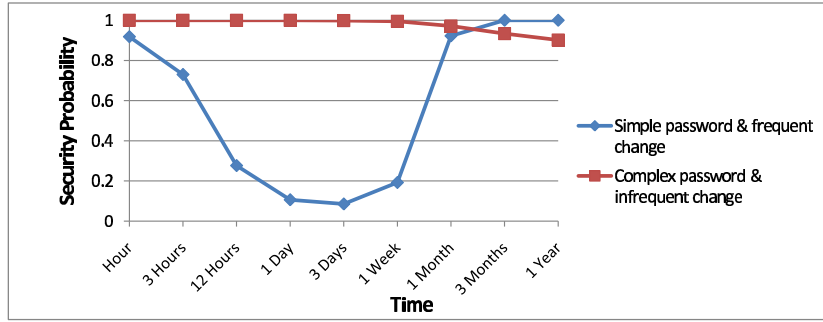
Figure 8. Password complexity vs. frequent maintenance trade-off.

The sixth experiment was designed to answer an important question: is it better to choose more secure passwords or to perform maintenance more frequently? The experiment considered two systems, one with six-character alphabetic passwords and once-a-week maintenance and the other one with eight-character alphanumeric passwords and maintenance every three months. The results are shown in Figure 8. Weak passwords with frequent maintenance are less secure in short term, but after a while (one year) complex passwords start to expire and the overall security (of the second system) decreases. Note also that changing passwords every week can be a huge administrative burden.
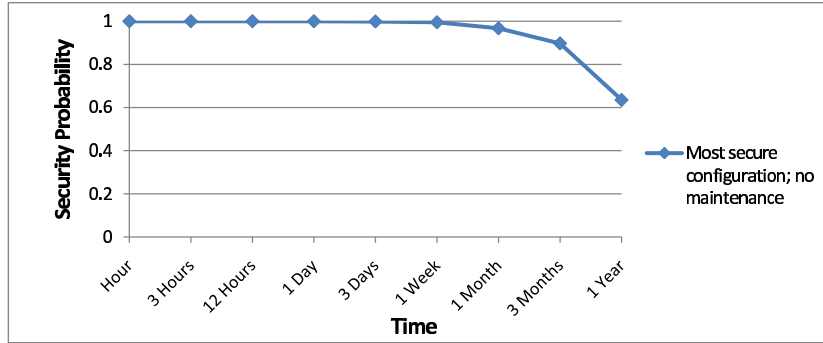

Figure 9. Impact of secure parameters with no maintenance.

The seventh experiment focused on a single configuration: the most secure configuration with no maintenance, complex (eight-character alphanumeric) passwords, ten system users and no malware. Figure 9 presents the results of the experiment. The importance of regular maintenance is clear – even a relatively secure configuration becomes less than 65% secure after one year without proper maintenance.

5. Security Recommendations

The simulation results provide valuable insight into securing VPNs, especially in industrial control environments where a tunnel is the only means to establish communications security and the tunnel may, therefore, last for a long period of time.

Based on the experimental results, the following recommendations can be made regarding VPN security:

- The aggressive mode for IPSec VPNs provides fast tunnel establishment and less overhead that render the mode an attractive option for industrial environments where timing is critical. However, the mode suffers from serious protocol flaws that can result in security breaches in a relatively short time. This mode should not be used in critical applications. Secure configurations using the main mode and certificate-based authentication provide stronger VPN tunnels at the expense of higher overhead and slower connection establishment.

- Long alphanumeric passwords or PSKs should be used to achieve acceptable security.

- Even with complex passwords, frequent maintenance must be performed to lower the risk of a successful attack, especially when the adversary has significant computational resources. Note that personal "supercomputers" and botnets can significantly reduce the password cracking time.

- A weak configuration can have a dominant effect even when malware infections are frequent. Securely configuring a VPN is the first step to countering attacks.

- Less populated VPNs are more secure. When a VPN has a large number of users, other parameters must be stronger (e.g., longer passwords and frequent maintenance). In the case of VPN tunnels for industrial control applications, it is advisable to keep the number of users as low as possible.

- Usernames (and IP addresses) used in a VPN must be changed or rotated periodically to reduce the risk of username enumeration attacks.

6. Related Work

Although probabilistic analysis has been widely used to investigate system reliability, its application to security has not attracted much attention until recently. Wang, *et al.* [28] have proposed the use of probabilistic models for analyzing system security. They have shown that modeling security using Markov chains can be quite informative and can facilitate the design of secure systems. Singh, *et al.* [27] have used probabilistic models to study the security of intrusion-tolerant replication systems.

Many previous efforts and standards recommend that VPNs be used to secure industrial control protocols. IEC 62351 [14–16] recommends the deployment of VPNs for protocols such as DNP3 and 61850. Okabe, *et al.* [21] propose the use of IPSec and KINK to secure non-IP-based control networks. Gungor and Lambert [9] discuss the use of MPLS and IPSec VPNs to provide security for electrical power system applications. Sempere, *et al.* [26] demonstrate the performance and benefits of using VPNs over IP (among other technologies) for supervisory control systems. Alsiherov and Kim [2] propose the use of IPSec VPNs to ensure integrity, authenticity and confidentiality in SCADA networks; however, they suggest that IPSec be configured in the PSK mode for efficient management. Patel, *et al.* [22] discuss the use of TLS and IPSec VPNs to wrap SCADA protocols. Alsiherov and Kim [1] suggest using IPSec between SCADA sites to provide security when IEC 62351 is not implemented.

Hills [12] has identified several VPN security flaws and has analyzed the presence of secure configurations in VPN deployments. Hamed, *et al.* [10] have developed a scheme for modeling and verifying IPSec and VPN security policies. Finally, Baukari and Aljane [4] have specified an auditing architecture for monitoring the security of VPNs.

7.   Conclusions

A stochastic model of a VPN and its environment provides a powerful framework for investigating the impact of various configurations and operational modes on VPN security in industrial control environments. Simulations of the model assist in quantifying the security of control protocols and evaluating security trade-offs, thereby providing a basis for the secure deployment of VPNs. The results also provide valuable recommendations for securely configuring VPNs in industrial control environments.

Our future research will study other VPN protocols (e.g., TLS and L2TP) and quantify their security properties. Also, we plan to incorporate detailed models of malware infections and man-in-the-middle attacks to study their impact more meticulously. Our research will also model other industrial control protocols using SANs with the goal of evaluating their benefits and limitations.

References

[1] F. Alsiherov and T. Kim, Research trend on secure SCADA network technology and methods, *WSEAS Transactions on Systems and Control*, vol. 8(5), pp. 635–645, 2010.

[2] F. Alsiherov and T. Kim, Secure SCADA network technology and methods, *Proceedings of the Twelfth WSEAS International Conference on Automatic Control, Modeling and Simulation*, pp. 434–438, 2010.

[3] G. Balbo, Introduction to stochastic Petri nets, in *Lectures on Formal Methods and Performance Analysis (LNCS 2090)*, E. Brinksma, H. Hermanns and J.-P. Katoen (Eds.), Springer Verlag, Berlin-Heidelberg, Germany, pp. 84–155, 2001.

[4] N. Baukari and A. Aljane, Security and auditing of VPN, *Proceedings of the Third International Workshop on Services in Distributed and Networked Environments*, pp. 132–138, 1996.

[5] M. Bellare, R. Canetti and H. Krawczyk, Keying hash functions for message authentication, *Proceedings of the Sixteenth International Cryptology Conference*, pp. 1–15, 1996.

[6] R. Brown, Stuxnet worm causes industry concern for security firms, Mass High Tech, Boston, Massachusetts (www.masshightech.com/stories /2010/10/18/daily19-Stuxnet-worm-causes-industry-concern-for-security-firms.html), October 19, 2010.

[7] D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. Doyle, W. Sanders and P. Webster, The Mobius framework and its implementation, *IEEE Transactions on Software Engineering*, vol. 28(10), pp. 956–969, 2002.

[8] S. Dispensa, How to reduce malware-induced security breaches, *eWeek .com*, March 31, 2010.

[9] V. Gungor and F. Lambert, A survey on communication networks for electric system automation, *Computer Networks*, vol. 50(7), pp. 877–897, 2006.

[10] H. Hamed, E. Al-Shaer and W. Marrero, Modeling and verification of IPSec and VPN security policies, *Proceedings of the Thirteenth IEEE International Conference on Network Protocols*, pp. 259–278, 2005.

[11] D. Harkins and D. Carrel, The Internet Key Exchange (IKE), RFC 2409, 1998.

[12] R. Hills, Common VPN Security Flaws, White Paper, NTA Monitor, Rochester, United Kingdom (www.nta-monitor.com/posts/2005/01/VPN-Flaws-Whitepaper.pdf), 2005.

[13] International Electrotechnical Commission, IEC 61850 Standard, Technical Specification IEC TS 61850, Geneva, Switzerland, 2003.

[14] International Electrotechnical Commission, Communication Network and System Security – Profiles including TCP/IP, Technical Specification IEC TS 62351-3, Geneva, Switzerland, 2007.

[15] International Electrotechnical Commission, Security for IEC 61850, Technical Specification IEC TS 62351-6, Geneva, Switzerland, 2007.

[16] International Electrotechnical Commission, Security for IEC 60870-5 and Derivatives, Technical Specification IEC TS 62351-5, Geneva, Switzerland, 2009.

[17] P. Li, W. Zhou and Y. Wang, Getting the real-time precise roundtrip time for stepping stone detection, *Proceedings of the Fourth International Conference on Network and System Security*, pp. 377–382, 2010.

[18] M. Majdalawieh, *Security Framework for DNP3 and SCADA*, VDM Verlag, Saarbruken, Germany, 2008.

[19] Modbus-IDA, Modbus Application Protocol Specification V.1.1b, Hopkinton, Massachusetts (www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf), 2006.

[20] N. Nayak and S. Ghosh, Different flavors of man-in-the-middle attack: Consequences and feasible solutions, *Proceedings of the Third IEEE International Conference on Computer Science and Information Technology*, pp. 491–495, 2010.

[21] N. Okabe, S. Sakane, K. Miyazawa, K. Kamada, A. Inoue and M. Ishiyama, Security architecture for control networks using IPSec and KINK, *Proceedings of the Symposium on Applications and the Internet*, pp. 414–420, 2005.

[22] S. Patel, G. Bhatt and J. Graham, Improving the cyber security of SCADA communication networks, *Communications of the ACM*, vol. 52(7), pp. 139–142, 2009.

[23] K. Paterson, A cryptographic tour of the IPSec standards, *Information Security Technical Report*, vol. 11(2), pp. 72–81, 2006.

[24] R. Pereira and S. Beaulieu, Extended Authentication within ISAKMP /Oakley (XAUTH), Internet Draft, 1999.

[25] W. Sanders and J. Meyer, Stochastic activity networks: Formal definitions and concepts, in *Lectures on Formal Methods and Performance Analysis (LNCS 2090)*, E. Brinksma, H. Hermanns and J.-P. Katoen (Eds.), Springer Verlag, Berlin-Heidelberg, Germany, pp. 315–343, 2001.

[26] V. Sempere, T. Albero and J. Silvestre, Analysis of communication alternatives in a heterogeneous network for a supervision and control system, *Computer Communications*, vol. 29(8), pp. 1133–1145, 2006.

[27] S. Singh, M. Cukier and W. Sanders, Probabilistic validation of an intrusion-tolerant replication system, *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 615–624, 2003.

[28] D. Wang, B. Madan and K. Trivedi, Security analysis of SITAR intrusion tolerance system, *Proceedings of the ACM Workshop on Survivable and Self-Regenerative Systems*, pp. 23–32, 2003.