Journal of Advance Computational Research
Vol. 3 No. 1 (January-June, 2018)

Quantitative Evaluation of Virtual Private Networks and its Implications for Communication Security in Industrial Protocols

Sanaz Rahimi and Mehdi Zargham
Department of Computer Science, Southern Illinois University Carbondale, 1000 Faner Drive, Carbondale IL

RECEIVED: November 015,2017. Revised March 16, 2018

Abstract: Virtual Private Networks (VPNs) are widely recommended to provide security for otherwise unsecured industrial and SCADA communication protocols. VPNs provide confidentiality, integrity, and availability and are often considered secure. However, implementation vulnerabilities and protocol flaws have exposed VPN weaknesses in many deployments. In this work, we use probabilistic modeling to evaluate and quantify the security of different VPN configurations. By simulating the VPN models in several experiments, we study the trade-offs and parameter dependence of each configuration. Using the evaluation results, we provide a few recommendations for secure VPN deployment in industrial systems.

I. INTRODUCTION

Virtual Private Network (VPN) is widely used today as a means of secure communication over an unsecured public network. VPNs provide security services such as confidentiality, integrity, and limited availability by forming encrypted tunnels between the communicating parties in a public network. VPN is recommended in the literature and by many critical infrastructure protection standards to secure the communication of process control, SCADA, and automation protocols [3], [5], [4], [21], [6], [22].

Although almost all these protocols are very reliable, they were not designed to resist malicious attacks. As a result, it has been proposed to wrap industrial protocols inside VPN tunnels to protect them from unauthorized access. For instance, industrial and SCADA protocols such as DNP3 [19], 61850 [1], and Modbus [2] are wrapped inside VPN tunnels in order to ensure their confidentiality, integrity, and availability [22]. However, little work has been done on the secure configurations necessary for a VPN tunnel or the maintenance required for its secure operations [22].

Although VPNs are trusted to provide secure communication in many cases, they are in fact attractive potential targets for attackers. First, VPNs carry sensitive information over a public network, so upon a successful break into a VPN tunnel, the attacker can maliciously alter sensitive industrial data or commands without physical access to the facility. Second, if other protection mechanisms such as strong access control are not deployed properly, the attacker can gain access to the internal SCADA systems through the VPN tunnel. Also, as industrial systems implement more security mechanisms, VPNs can become the weakest link in their security chain.

VPNs have known security vulnerabilities. As studies shows [17], most of the VPN implementations suffer from serious security flaws which can be easily exploited by an attacker to fabricate, intercept, modify, or interrupt traffic. Some of these vulnerabilities are implementation specific; i.e., they are flaws in the specific implementation of the protocol because of bad coding, incomplete implementation, or bad implementation choices for conditions not specified in the standard. More seriously, VPN has other vulnerabilities in the underlying protocols which cannot be avoided by good implementation. Finally, as recent incidents have shown, sophisticated malware attacks [11] can modify the configurations of a control system (including VPNs) stealthily, damaging their operations. The solutions to these problems are proper configurations, regular maintenance, and configuration validation which can only be performed correctly if the administrators fully understand the internal details of the protocol.

In this paper, we first provide an overview of VPNs, their vulnerabilities, and important industrial protocols which use (or recommend the usage of) VPN for secure communication. We then model VPNs using Stochastic Activity Networks [25] (an extended form

Correspondence email address: **srahimi,mehdi@cs.siu.edu**

of Petri nets [8]) and analyze the probability of a successful breach against different configurations and parameters. Based on the findings, we provide a set of recommendations for secure deployment of VPN in industrial systems.

Our contributions are as follows:

- We provide a probabilistic model of a security protocol (VPN) and solve the model using simulation.
- We quantify the security of VPNs for different choices of parameters including the key length, the mode of operation, the number of users in the system, and the maintenance frequency. To the best of our knowledge, we are the first to quantify the security of VPNs.
- Based on the simulation results, we provide a few recommendations for secure deployment of VPNs in industrial systems.

The rest of the paper is organized as follow. Section II describes the VPN protocol and its vulnerabilities. Section IV describes the probabilistic model used to represent the security of VPNs and explains the details involved in studying such a model. The experimental results for different configurations and trade-offs are presented in Section V. Section VI provides a few recommendations for secure VPN deployment in industrial systems. We review the related work in Section VII before concluding the paper in Section VIII.

II. REVIEW OF VPN AND VULNERABILITIES

VPNs are categorized with respect to their layer: transport layer (SSL), network layer (IPSec), and link layer (L2TP). In this work we study IPSec VPNs. IPSec [23], as one of the underlying VPN protocols 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 (tunnel vs. transport). Detailed description of IPSec is beyond the scope of this work; here, we only briefly describe those features of the protocol which are necessary for our discussion. The reader may refer to the IPSec RFC [23] for more information.

A. IPSec

IPSec provides security services via Authentication Header (AH) and Encapsulation Security Payload (ESP) protocols. ESP encrypts the packet payload and some fields of the IP header providing confidentiality and limited integrity. It adds ESP header and trailer to the IP packet. AH, on the other hand, provides integrity by adding the HMAC [10] of the entire packet (payload and full IP header.) It is important to note that AH does not provide confidentiality because it leaves the packet in plaintext.

IPSec can operate in two different modes: tunnel or transport. Transport mode is used when end-to-end security is desired and both end nodes support IPSec. In this mode, the original IP header is preserved for routing purposes and the ESP/AH headers are added under the IP header. Tunnel mode, on the other hand, is used when either the end machines do not support IPSec or 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 is 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). For a bidirectional communication, two SAs are required.

IPSec uses Internet Key Exchange (IKE) protocol to manage and exchange encryption keys and algorithms. IKE itself 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 it does not define the specific algorithms or keys. IKE uses Oakley and SKEME protocols for the actual key exchange and agreement on the acceptable cryptographic algorithms.

IKE is an important protocol in establishing VPN channels as many of the 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 here. IKE has three important modes: main mode, aggressive mode, and quick mode.

Main mode is used for authentication and key exchange in the initial phase of IKE. This phase

assumes that no SA is present and the two parties would like to establish SAs for the first time. It includes 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. Finally, the last two messages authenticate peers using signatures or certificates. Note that the identities of the peers in the main mode are often their IP addresses.

Similarly, the aggressive mode is used for the initial key exchange, but it is faster and more compact than the main mode. It involves a total of three messages containing the main mode parameters presented in a more compact form. Key and policy exchange is performed in 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; hence opening this mode of operations to a realm of vulnerabilities.

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

IKE authenticates peers using three different methods: Pre-Shared Key (PSK), public key encryption, or digital signature. In the PSK method which is the traditional username/password authentication, the peers share a secret through a back channel and exchange the hash of its value for authentication. Unfortunately, although this method has known vulnerabilities, it is the only mandatory authentication method according to the RFC [16]. Public key encryption is another method of authentication in which the peers digitally sign their identities. Public keys should be provided by other means beforehand. Digital certificates can also be used for authentication in IKE. In this mode, the peers exchange certificates to mutually authenticate themselves.

B. VPN Vulnerabilities

VPNs have known flaws and vulnerabilities. Username enumeration is one of the important vulnerabilities present in many VPN implementations. Username enumeration vulnerability refers to an implementation flaw in which the username/password authentication mechanism responds to invalid username and password differently. As a result, an attacker can find a list of valid usernames which can be used later to discover their passwords.

When using IKE 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 containing the hash of the identity and the PSK (password). Many implementations of VPN in response to a wrong username, send an error message, send a packet with NULL PSK, or do not respond at all. As a result, an attacker can conclude whether a username exist or not by sending a single packet (enumerate username). Upon finding the first username, attacker can perform efficient guessing by generating other usernames with the same pattern as the one it found (e.g. first letter of the first name concatenated with the last name); thus finding other usernames faster. If VPN is used in the main mode, the identity is an IP address, not a username.

Hill [17] proposes that a secure VPN implementation can return the hash of a randomly generated password each time it receives an invalid username. This does not solve the problem because an attacker can still send two different packets with the same username and if it gets two different hashes back, it knows that the username does not exist and vice versa. Furthermore, attacker can delay these two packets with arbitrary number of packets for other usernames to flush any buffer that the server may use to keep track of 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 send back the hash of this value. In this scenario, the server always responds to a username with a unique hash value thus masking any such attacks.

As soon as the attacker finds a valid username, it can receive the hash of its password from the server (using in aggressive mode PSK). The attacker can then perform an offline password cracking on the hashed value to obtain the password. Note that because the hashed probabilistic model of VPN password is not encrypted, offline cracking can be very fast. This poses a serious threat especially to short passwords. Even if IKE operates in main mode PSK, this attack is still

possible if the attacker can perform a man-in-the-middle (MITM) 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.

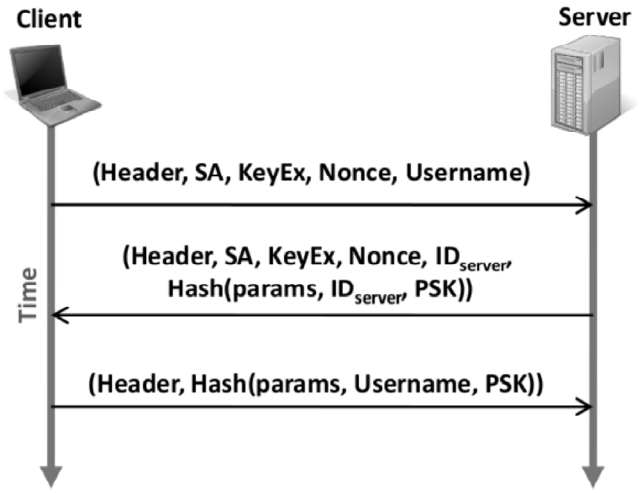

Figure 1: The IKE Aggressive Mode with PSK

Figure 1 illustrates the IKE protocol exchanges in the aggressive-PSK mode. The header, SA, KeyEX, and Nonce denote the IKE header parameters, security association, key exchange parameters (e.g. Diffie-Hellman parameters), and a random nonce respectively. The Username and IDserver denote the client's and server's identities. Note that the client knows all the hash input values except the PSK (params and IDserver), so it effectively possesses the hash value of the PSK after the second message. The params argument shows the other exchanged parameters, not important in this work.

When a username/password pair is successfully found, the first phase of the 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 phase two of IKE, but this phase is vulnerable to MITM attack as suggested in the standard [24]. The reason for such vulnerability is that XAUTH must be used in conjunction with the first phase of IKE; if this phase is not performed properly, there will be no security guarantee from XAUTH. Hence, an attacker would be able to authenticate successfully if he can perform a MITM attack.

Other vulnerabilities of different VPN implementations include: VPN fingerprinting (inferring information about the device from its behavior), unsecured storage of passwords (e.g. in registry, plaintext in memory, etc.), lack of account lockout, poor default configurations, and unauthorized modification of configurations. However, for our discussion, we do not consider these vulnerabilities because they are implementation specific, and in some cases they require other exploitable flaws for a successful attack (e.g. unsecured storage of password requires a malware to exploit it).

III. MODELING METHODOLOGY

We have to use a modeling abstraction to study the probabilistic cracking of a VPN connection. The model can then be solved quantitatively using simulation. We could have developed our own simulator to study VPNs, but because there are both rich abstractions and powerful simulator available, we decided to use them instead. We model the security of VPN using an extension of Petri nets [8] called Stochastic Activity Networks (SANs) [25]. The Mobius [12] tool is used to specify the model and to find its numerical solution. For the sake of completeness, we provide a brief description of SANs.

(A) Stochastic Activity Networks

A SAN is a Petri net with a few simple extensions. In its simplest form a SAN consists of places and activities (Figure 2a). Each place can have a number of tokens (a.k.a. its marking) and each activity moves tokens from one place into another. In Figure 2a, Activity1 moves tokens from Place1 to Place2. An activity can move tokens either at fixed times (deterministic) or at random times (probabilistic). A deterministic activity can be specified with its rate whereas a probabilistic activity is specified with its probability distribution function (PDF). For example, an activity can transfer tokens at the rate of one token per second or at random times distributed exponentially. Exhaustive search in a space of size N used extensively in this work has a uniform distribution with a PDF as follows.

$$f(x) = \begin{cases} \frac{1}{N} \; if \;\; 0 < x < N \\ 0 \qquad otherwise \end{cases}$$

In addition to Petri net abstractions, a SAN can also include an input gate (Figure 2b). An input gate

can define an arbitrary condition (predicate) for enabling an activity. For example, if Input_Gate1 in the figure has the following predicate, Activity1 is only enabled when Place1 has more tokens than Place2.

```
if(Place1->Mark() > Place2->Mark())
return 1;
else
return 0;
```

Note that the ->Mark() operator is a pseudo operator that returns the marking of a place.

Moreover, a SAN can also include an output gate. An output gate defines an arbitrary effect after an activity has been enabled (Figure 2c). For example, if Output_Gate1 in the figure has the following function, whenever Activity1 is enabled it takes one token from Place4 and puts one token in Place2 and one in Place3.

```
Place2->Mark()++;
Place3->Mark()++;
Place4->Mark()—;
```

IV. VPN MODEL

In this section, we describe a probabilistic model for the security of VPN. Using probabilistic modeling, we quantify the security of an important protocol and make suggestions for its secure implementation, configuration, and operation. According to a study by Hill [17], VPN tunnels are misconfigured in most cases (~ 90%).

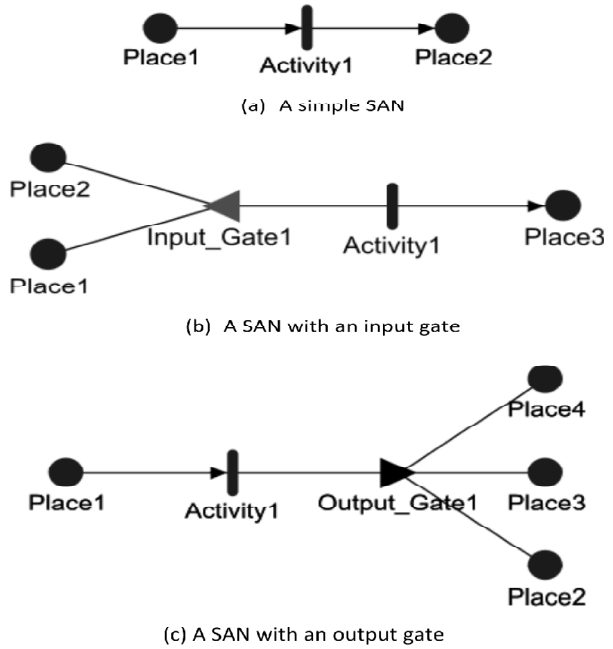

Figure 2: Simple SAN Abstractions

We explain the details of the SAN model and different choices of the parameters. All times in the model are expressed in minutes. The model consists of 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 [12].

The first atomic model (ike) models the weaknesses of the protocol (Figure 3). A global variable determines whether the VPN is working in aggressive mode or main mode. If it is configured in the aggressive mode, an activity models username enumeration attack. We consider usernames which consists of alphabetic characters and have a length of at most six characters. The total number of possible usernames is approximately 309 million. If the round-trip-time (RTT) between the scanner and the server is in the order of tens of milliseconds [18] and a window of 10 packets is used, on average it takes 1 ms to check for each username, so we assume that 1000 usernames can be checked per second. If the RTT is larger, the window can be chosen accordingly to achieve this rate. With the given rate, it is possible to exhaustively scan the username space in ~3.5 days. Sophisticated attackers can do better if they have a fast connection to the server or they use intelligent guessing algorithms once they find a username; however we consider a simple attacker to find an upper bound on the security of VPN. Note that since username scanning does not typically cause account lockout, this process will not be stopped by the server.

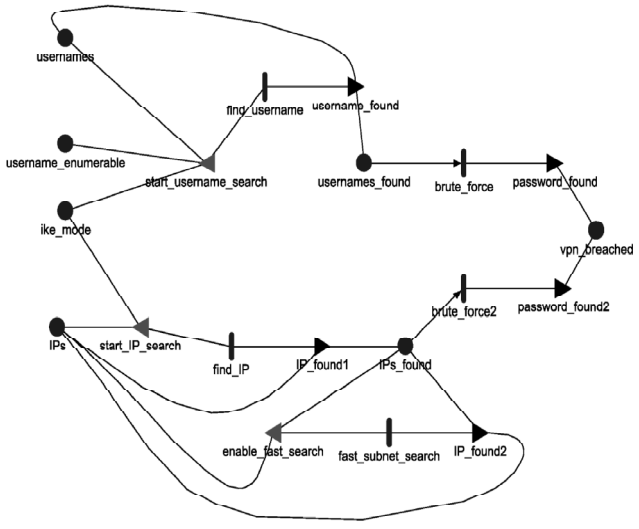

Figure 3: The Probabilistic Model of VPN

The rate of username enumeration is also proportional to the number of users in the system (more valid users result in a faster enumeration using exhaustive search). This is modeled by multiplying the base rate (1 per 3.5 day = 1.94E-4 per min) 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 usernames to "usernames_found" using the output gate "username_found." As soon as a username is found, the attacker starts offline attack to find its 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 (~1.9E+7 attempts per min) [17]. The cracking speed also depends heavily on password complexity. We run the model using different password space sizes to account for this fact. Table I illustrates the password space size for different password complexities.

The rate of successful attempts is also 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) have negligible time compared to brute force or username enumeration. As result, after a username/password pair is found a token is placed in "vpn_breached."

Table I
Password Space Size for Different Complexities

Complexity	*Space Size*
6 characters a-z	3.1E+8
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

The other possible mode of operation is the main mode. As mentioned before, in this mode, the identities are IP addresses of the peers. The space of the 32-bit IP addresses is approximately 14 times larger than the space of six-character usernames; thus the "find_IP" activity which randomly searches the IP address space has a base rate which is one fourteenth of that of "find_username." However, upon finding a valid IP address, the attacker can perform subnet based search, so finding other IP addresses is much faster. We assume that most of the clients have the same subnets making this fast subnet-based search possible. Note that for this mode to be enabled, at least one IP must be found by random searching. Upon finding a valid IP address, the attacker exhaustively searches the space of pre-shared keys (PSKs) similar to the aggressive mode, placing a token in "vpn_breached" whenever an IP/PSK pair is found.

The second atomic model in the SAN, models the behavior of the VPN environment and its operational maintenance (Figure 4). VPNs are specifically vulnerable to malware attacks [13]. Malwares can modify the configuration of a VPN tunnel stealthily. Since the VPN tunnel remains operational after the modification, it is difficult for the administrator to detect such an attack.

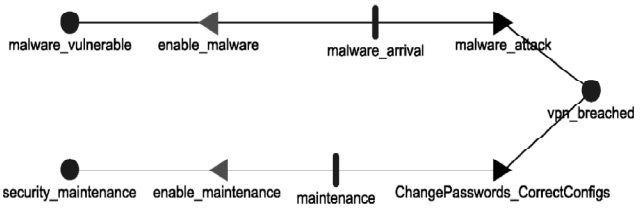

Figure 4: VPN Malware Infection and Maintenance Models

We model two different types of environment: one in which malware attacks exist and another one without malwares. A malware can modify the VPN configuration for it to send packet in plaintext, so arrival of a malware is synonymous to the VPN breach.

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

Maintenance of the VPN by the administrator is a preventive and/or corrective action that can improve security. Maintenance includes changing passwords and checking for bad configurations. If the VPN configuration is modified by a malware, maintenance operation can secure the VPN by correcting the configuration and patching for that specific malware. Also regularly changing passwords can mitigate exhaustive search attacks, making the VPN secure. On the other hand, password changes do not affect the usernames/IP enumeration, so in the model, that

activity does not flush "usernames_found" or "IPs_found" places.

V. RESULTS

We present the results from different experiments on the SAN model described in the previous section. Our goal is 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 is studied at different times: one hour, three hours, twelve hours, one day, three days, one week, one month, three months, and one year.

The security of VPN is studied for different modes of IKE (aggressive vs. main mode), password complexities (Table I), number of users/machines (1, 10, 100, and 250), environments (with or without malware attacks), and maintenance rates (once every week, month, three months, year, and no maintenance.) As the effects of many different factors are being studied, a large number of experiments are possible – not all of which are meaningful. We only study the interesting and meaningful experiments in which a few parameters change at a time.

The first experiment compares the security of the aggressive mode and main mode. The main mode is generally more secure because for offline password cracking, the attacker has to perform a MITM attack successfully. Even if the attacker can always perform a MITM attack, the space of 32-bit IP addresses is larger than the space of usernames. In the experiment, it is assumed that an attacker can perform a MITM attack; otherwise, the main mode is not vulnerable to the attack (i.e. the probability of security stays at 1 at all times.)

The passwords (or PSKs) for both modes are chosen from the space of six alphabetic characters (3.1E+8), the system has 10 users (usernames or IP addresses), and no security maintenance or malware is present. The results are illustrated in Figure 5. As the figure shows, the security of a VPN tunnel diminishes over time. In the aggressive mode, the security declines faster than the main mode. The aggressive mode is less than 50% secure after 6 hours whereas the main mode declines to that level after about 4 days. Notice the short lifetime of a six-character alphabetic password for a VPN tunnel.

The second experiment studies the effect of different password complexities on the overall security of VPN (aggressive mode). To observe the effect of password complexity alone, maintenance and malware attacks are turned off for this experiment. The system has 10 different users. The results are illustrated in Figure 6.

Intuitively, the overall security of VPN increases with password complexity. Note that eight-character alphanumeric passwords are secure over a much larger period of time, but even this type of password is less than 65% secure after a year. Six-character alphanumeric passwords are less than 60% secure after just one day.

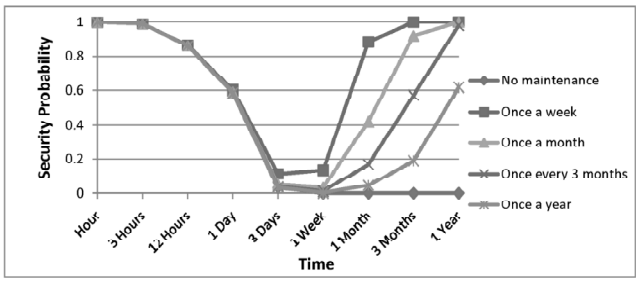

Figure 5: Security of Aggressive Mode versus Main Mode

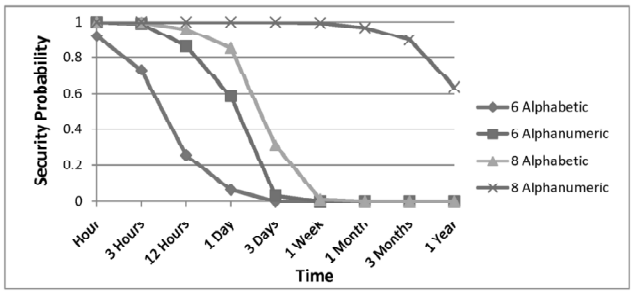

Figure 6: The Effect of Different Password Complexities on VPN Security of VPN

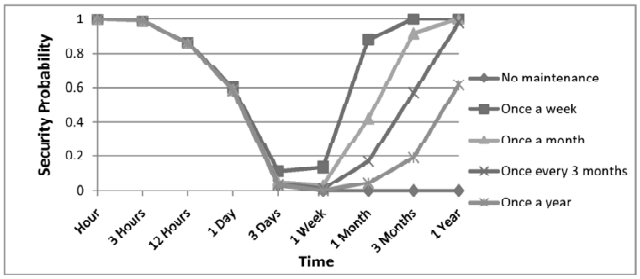

Figure 7: The Effect of Different Maintenance Frequencies on VPN Security

The next experiment studies the effect of different maintenance frequencies on the security. For this experiment, IKE works in the aggressive mode, there are 10 users in the system, and passwords are six-

character alphanumeric. The results in Figure 7 illustrate that frequent maintenance can mitigate the effect of weak configurations. Note that the probability of security declines with time until it reaches a minimum before any maintenance starts. After that it increases faster with more frequent maintenance. Since the rate of maintenance is higher than the rate of password cracking in each case, the security reaches 1 in the steady state solution. This does not mean that it is impossible to break the VPN tunnel as time grows; it simply implies that the portion of time that the VPN tunnel is breaches diminishes over longer time periods.

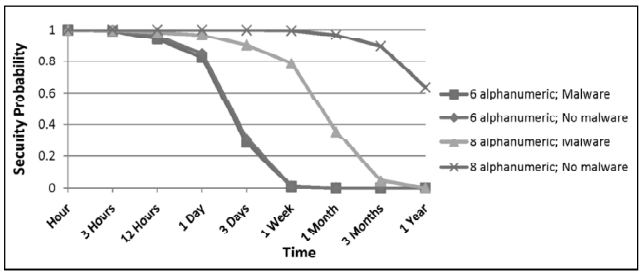

Figure 8: The Effect of Malwares on VPN Security

The fourth experiment studies the effect of malware attacks vs. weak passwords on the VPN. Two password complexities (6-and 8-character alphanumeric) are plotted in Figure 8 with and without frequent malware attack present (once a month.) For this experiment, IKE uses the aggressive mode, and no maintenance is provided. A counter-intuitive result from this 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 the attacker to crack a six-character alphanumeric password. For a strong password, on the other hand, frequent malware infections can considerably weaken the VPN security. We conclude that the impact of malware infection depends on the configuration of the VPN. If the rate of password cracking is higher than that of infection, malwares will have little impact on the system. As a result, the effort must first go into securely configuring the VPN, then into malware protection. Note that this study only considers the effect of malware on the security of the VPN tunnel. Malware infections may have many more negative security impacts which are not modeled here.

Next, we study the effect of the number of users in the system on its overall security. As illustrated in Figure 9, populated systems are far less secure than systems with a few users because an attacker has higher chances of finding valid usernames/passwords (or IPs/ PSKs.) For this experiment, IKE is in aggressive mode, the passwords are six-character alphanumeric, and no maintenance or malware attack is present.

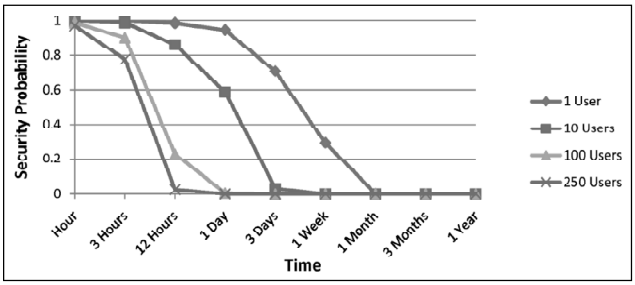

Figure 9: The Effect of the Number of Users on VPN Security

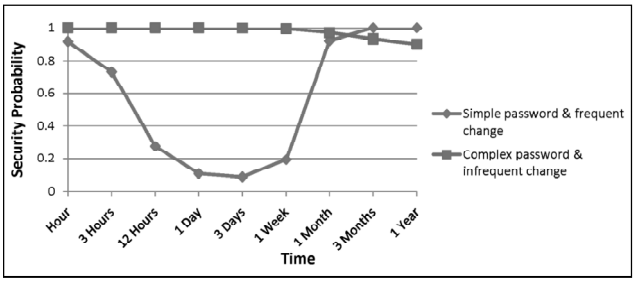

Figure 10: Password Complexity vs. Frequent Maintenance Trade-off

The next experiment answers an important question: is it better to choose more secure passwords or to perform maintenance more frequently? For this experiment, we consider two different systems: one with six character alphabetic passwords and once a week maintenance, the other one with 8-character alphanumeric passwords and maintenance every three months. The results are illustrated in Figure 10. Weak passwords with frequent maintenance are less secure in short term, but after a while (1 year) complex passwords start to expire and the overall security of the latter system diminishes. Note that changing passwords every week can be a huge administrative burden.

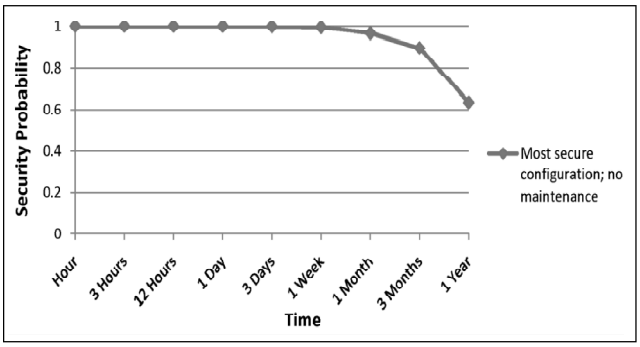

Figure 11: The Effect of Secure Parameters with no Maintenance

The last experiment focuses on a single configuration: the most secure configuration, but no maintenance; i.e. passwords are complex (8-character alphanumeric), no malware is present, and only ten users exist in the system. Figure 11 illustrates the results. As can be seen from the figure, even a relatively secure configuration is less than 65% secure after a year without proper maintenance.

VI. RECOMMENDATIONS

The simulation results from our model of VPNs provide insight into securing VPNs especially for industrial applications where a tunnel may last a longer period of time and it is the only means of communication security. We provide the following recommendations:

The aggressive mode of IPSec VPNs provide fast tunnel establishment and less overhead which may make it an attractive choice for industrial protocols where timing is critical. It, however, suffers from serious protocol flaws that can result in a breach in relatively short time. This mode must be avoided for critical applications as much as possible. Secure configurations using the main mode and certificate-based authentication provide stronger VPN tunnels at the expense of higher overhead and slower connection establishment.

Passwords or PSKs must be long and alphanumeric to achieve acceptable security.

Even with complex passwords, frequent maintenance must be performed to lower the risk of a successful attack, especially when facing an adversary with a large computational power.

Weak configurations can have the dominant effect even when malware infections are frequent. Securely configuring a VPN must be the first step in countering cyber attacks.

Less populated VPN systems are more secure. When the system has a large number of users, other parameters in the system must be stronger (longer passwords, more frequent maintenance). Specifically in industrial tunnels, it is advisable to keep the number of valid users as low as possible.

The usernames (IP addresses) used in a VPN system must be changed or rotated after a while to mitigate the risk of username enumeration attacks.

VII. RELATED WORK

Although probabilistic analysis has been used for studying reliability of systems, its application to security has not attracted much attention until recently. Wang *et al.* [28] propose the use of probabilistic models for analyzing the security of systems. Singh *et al.* [27] use probabilistic models to study the security of intrusion tolerant replication systems. Wang et al. [28] build a general model for security of a system using Markov chains. It also discusses the feasibility of modeling for security purposes. They show that not only it is possible to model security using Markov chains, but also the models can be quite informative and can facilitate designing secure systems.

Many previous efforts and standards suggest using VPNs to secure industrial protocols. The IEC standard suite 62351 [3], [5], [4] recommends the deployment of VPN to secure industrial protocols such as DNP3 and 61850. Okabe *et al.* [21] propose using IPSec and KINK to secure non-IP based control networks. Gungor and Lambert [14] discuss MPLS and IPSec VPNs to provide security in electric system automation. Sempere *et al.* [26] evaluate the performance and cost of VPN over IP (among other technologies) for supervision and control systems of a purification network. Alsiherov and Kim [7] propose using IPSec VPN to ensure integrity, authenticity, and confidentiality of SCADA networks. Unfortunately it has been suggested for IPSec to be configured in pre-shared key mode for efficient management. Patel *et al.* [22] discuss using TLS or IPSec VPNs to wrap SCADA protocols. Alsiherov and Kim [6] suggest using IPSec between SCADA sites to provide security if 62351 is not implemented.

Hill [17] proposes VPN security flaws and surveys compliance with secure configurations in VPN deployments. Hamed *et al.* [15] provide a scheme to model and verify IPSec and VPN security policies. Finally, Baukari and Aljane [9] describe an auditing architecture to monitor the security of VPNs.

VIII. CONCLUSION

In this work, we have customized the application of probabilistic security modeling to model the security of the VPN protocol. VPN is proposed in the literature and standards to secure communication over public networks for industrial and SCADA protocols.

We built a stochastic model of VPN and its environment and investigated the effect of different

configurations and operational modes on its security. By simulating the model, we quantified the security of the protocol and studied different trade-offs, thus providing a basis for secure deployment of VPN. Using the insight from the results, we provided a list of recommendations for secure configuration of VPN in industrial systems.

Future work can study other VPN protocols (TLS, L2TP) and quantify their security properties. Also we plan to incorporate more detailed models for malware and MITM attacks, so that their impact can be studied more meticulously. Moreover, we plan to model other security and industrial protocols using Stochastic Activity Networks to evaluate their properties and limitations. We also leave testbed implementation of the attacks discussed in this paper for future work. The implementation can provide a framework for cross-validation of the simulation results

References

[1] IEC 61850 Standard. Technical Specification IEC TS 61850, IEC, August 2003. *http://www.iec.ch/*.

[2] MODBUS Protocol Specification. Specification V1.1b, Modbus, December 2006. *http://www.modbus.org/docs/Modbus Application Protocol V1 1b.pdf*.

[3] Communication Network and System Security– Profiles Including TCP/IP. Technical Specification IEC TS 62351-3, IEC, June 2007.

[4] Security for IEC 61850. Technical Specification IEC TS 62351-6, IEC, June 2007.

[5] Security for IEC 60870-5 and Derivatives. Technical Specification IEC TS 62351-5, IEC, June 2009.

[6] F. Alsiherov and T. Kim. Research Trend on Secure Scada Network Technology and Methods. *WSEAS Trans. Sys. Ctrl.*, 5:635–645, August 2010.

[7] F. Alsiherov and T. Kim. Secure scada network technology and methods. In Proceedings of the 12th WSEAS International Conference on Automatic Control, Modelling & Simulation, ACMOS'10, pages 434–438, Stevens Point, Wisconsin, USA, 2010. World Scientific and Engineering Academy and Society (WSEAS).

[8] G. Balbo. Introducation to Stochastic Petri Nets, pages 84–155. Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[9] N. Baukari and A. Aljane. Security and auditing of vpn. Services in Distributed and Networked Environments, Workshop, 0:132, 1996.

[10] M. Bellare, R. Canetti, and H. Krawczyk. Keying Hash Functions for Message Authentication. In Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '96, pages 1–15, London, UK, 1996. Springer-Verlag.

[11] R. H. Brown. Stuxnet Worm Causes Industry Concern for Security Firms, October 2010. *http://www.masshightech.com/stories/2010/10/18/daily19-Stuxnet-worm-causes-industry-concern-for-security-firms.html*.

[12] D. D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. G. Webster. The Mobius Framework and its Implementation. *IEEE Trans. Softw. Eng.*, 28:956–969, 2002.

[13] S. Dispensa. How to reduce malware-induced security breaches. eWeek, March 2010.

[14] V. C. Gungor and F. C. Lambert. A Survey on Communication Networks for Electric System Automation. *Comput. Netw.*, 50:877–897, 2006.

[15] H. Hamed, E. Al-shaer, and W. Marrero. Modeling and Verification of IPSEC and VPN Security Policies. In in Proc. IEEE Int. Conf. Netw. Protocols, 2005.

[16] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409, Internet Engineering Task Force, 1998.

[17] R. Hills. Common vpn security flaws. White paper, NTA Monitor Ltd., January 2005. *www.ntamonitor.com/posts/2005/01/VPN-Flaws-Whitepaper.pdf*.

[18] P. Li, W. Zhou, and Y. Wang. Getting the Real-time Precise Round-trip Time for Stepping Stone Detection. Network and System Security, International Conference on, 0:377–382, 2010.

[19] M. Majdalawieh. Security Framework for DNP3 and SCADA. VDM Verlag, Saarbrücken, Germany, Germany, 2008.

[20] G. Nath Nayak and S. Ghosh Samaddar. Different flavours of man-in-the-middle attack, consequences and feasible solutions. In Proceedings of the Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, pages 491 – 495, Chengdu, 2010.

[21] N. Okabe, S. Sakane, K. Miyazawa, K. Kamada, A. Inoue, and M. Ishiyama. Security architecture for control networks using ipsec and kink. In Applications and the Internet, 2005. Proceedings. The 2005 Symposium on, 2005.

[22] S. C. Patel, G. D. Bhatt, and J. H. Graham. Improving the Cyber Security of Scada Communication Networks. Commun. ACM, 52:139–142, 2009.

[23] K. G. Paterson. A cryptographic tour of the IPSEC Standards. *Inf. Secur. Tech. Rep.*, 11:72–81, 2006.

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

[25] W. H. Sanders and J. F. Meyer. Stochastic Activity Networks: Formal Definitions and Concepts, pages 315–343. Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[26] V. Sempere, T. Albero, and J. Silvestre. Analysis of Communication Alternatives in a Heterogeneous Network for a Supervision and Control System. *Comput. Commun.*, 29:1133–1145, May 2006.

[27] S. Singh, M. Cukier, W. H. Sanders, and W. H. S. Probabilistic Validation of an Intrusion-tolerant Replication System. In Proc. International Conference on Dependable Systems and Networks (DSN 2003, pages 615–624, 2003.

[28] D. Wang, B. B. Madan, and K. S. Trivedi. Security Analysis of Sitar Intrusion Tolerance System. In Proceedings of the 2003 ACM Workshop on Survivable and Self-regenerative Systems: in Association with 10th ACM Conference on Computer and Communications Security, SSRS '03, pages 23–32, New York, NY, USA, 2003. ACM.