DETECTANDO PROBLEMAS EM REDES INDUSTRIAIS ATRAVÉS DE MONITORAMENTO CONTÍNUO

Jan Seidl [1]

Marcelo Ayres Branquinho [2]

RESUMO

Redes de automação oferecem uma gama de aplicações e dados em tempo real, tornando-se necessário o monitoramento contínuo da qualidade dos serviços (QoS). Os parâmetros do QoS (*Quality of Service*) procuram atender prioridades como alocação de largura de banda e controle de latência da rede. Existem vários parâmetros de QoS para caracterizar uma rede de computadores e que podem ser utilizados com propósitos de monitoramento.

Cada rede SCADA, em um estado saudável, apresenta um QoS específico, que raramente muda, dado o processo repetitivo das operações de IACS (*Industrial Automation and Control Systems*). O monitoramento contínuo dos parâmetros de QoS de uma rede de automação pode antecipar problemas como a contaminação por malware e falhas em equipamentos como switches e roteadores. É muito importante estar ciente das mudanças de comportamento, a fim de receber alertas e prontamente lidar com eles, evitando incidentes que possam comprometer o funcionamento da rede e ser financeiramente ou ambientalmente custosos.

Além do monitoramento do tráfego de rede, também é necessário monitorar o consumo de recursos de servidores críticos, tais como o processamento (CPU), memória, capacidade de armazenamento, falhas no disco rígido, entre outros.

Este trabalho tem como objetivo estabelecer um método pelo qual os profissionais de segurança SCADA poderão diferenciar e qualificar quaisquer problemas que possam estar ocorrendo através do monitoramento contínuo de parâmetros da rede de automação, dando uma abordagem mais comportamental do que as atuais, baseando-se em assinaturas de desempenho. Apresentamos uma série de testes realizados em nossos laboratórios, a fim de medir o desempenho de alguns parâmetros de uma rede de automação simulada, utilizando uma pequena *sandbox* de uma rede SCADA. Primeiro medimos os parâmetros normais da rede e colhemos os seus principais gráficos obtidos com as ferramentas adequadas. Em uma segunda etapa, praticamos diversos ataques contra a rede de automação simulada. Durante todos os ataques foram coletados os parâmetros operacionais da rede e seus principais gráficos.

Na conclusão do trabalho, comparamos os gráficos da rede no estado saudável com os gráficos da rede com os incidentes de segurança descritos acima. Detalhamos como os parâmetros de rede foram afetados por cada tipo de incidente e construímos uma tabela que mostra a forma como os principais parâmetros de uma rede de automação foram afetados pelos ataques.

Palavras-chave: Monitoramento, SCADA, Segurança, *Malware,* Ataques.

1 CTO da TI Seguro Segurança da Informação Ltda, Brasil (http://br.linkedin.com/in/janseidl)

2 *CEO em TI Segurança Segurança da Informação Ltda, Brasil (http://br.linkedin.com/in/marcelobranquinho)*

1 SOBRE O MONITORAMENTO DE REDES DE AUTOMAÇÃO

O Monitoramento de rede de automação é o termo usado para descrever um processo que monitora continuamente uma rede de automação e notifica o administrador de rede quando um dispositivo falha ou uma parada de funcionamento ocorre. Esta notificação é feita normalmente através de sistemas de mensagens (geralmente e-mail e *traps* SNMP). O monitoramento de rede é normalmente feito através do uso de aplicações dedicadas e/ou ferramentas comerciais. O comando ping, por exemplo, é um tipo de ferramenta de monitoramento de rede.

1.1 ATIVOS A SEREM MONITORADOS

Para monitorar um sistema de controle, nós precisamos primeiramente saber exatamente quais os dispositivos que existem na rede e como eles se comunicam uns com os outros. Quase todas as peças de hardware de rede em uma planta industrial podem ser monitoradas. Desde servidores SCADA, estações de supervisão/controle a CLPs, inúmeros itens podem ser monitorados e nos ajudar na prevenção e resposta rápida a incidentes.

2 PREPARANDO O AMBIENTE DE MONITORAMENTO

Os monitoramento de dados pode ser muito intenso e frequente. Dado este axioma, é altamente recomendável que seja criado um segmento de rede totalmente separado, destinado exclusivamente ao monitoramento. Isso vai evitar que os dados interfiram com o legítimo controle/fiscalização do tráfego no nível de rede e pode ajudar no isolamento do tráfego contra *sniffing* e outros ataques.
A quantidade adequada de servidores deve ser configurada de acordo com o número de ativos e locais a serem monitorados. A maioria das soluções pode operar em modo de alta disponibilidade e em *cluster* de alto desempenho. É importante determinar o processamento e capacidade da rede do ativo para determinar quando deverá ser usada uma abordagem com agente ou se um monitoramento passivo (monitoramento através de ping ou SNMP) deve ser usado.
Tenha em mente que as soluções de monitoramento geralmente são acompanhadas de uma solução de *back-end* e, por questões de performance, bancos de dados nunca devem partilhar o seu disco rígido com dados de outra aplicação.
Escrever uma matriz de tráfego industrial também é recomendado para determinar visualmente que ativos precisam se comunicar com outros ativos, e através de que códigos de função, para que possamos aprimorar as *triggers* de monitoramento.

Abaixo está um exemplo de uma matriz de tráfego industrial:

Origem	Destino	Códigos de Função
192.168.1.15	192.168.1.1	3, 16
192.168.1.18	192.168.1.1	3

Tabela 1: Exemplo de uma matriz de fluxo de tráfego industrial

2.1 MONITORANDO A SAÚDE DA MÁQUINA

O monitoramento da saúde de uma máquina pode ajudar na prevenção de problemas que poderiam interromper o funcionamento de programas e afetar as operações de supervisão ou controle. Com o monitoramento de desempenho ativo, problemas podem ser previstos e resolvidos antes que eles aconteçam.
Itens comuns são monitorados como espaço livre/usado em disco (aplicações podem falhar se não puderem gravar arquivos temporários), *Disk I/O* (pode indicar falta de memória [paginação] ou roubo de dados), usuários logados, número de tentativas de login sem sucesso (pode indicar comprometimento do sistema), número de conexões de entrada, número de conexões de saída, taxa de entrada/saída de pacotes (pode indicar a roubo de dados ou conexões ilegais ou mesmo algum malware), CPU e uso de memória do sistema (pode indicar *worms* e *rootkits*).

2.2 MONITORANDO ERROS NO SISTEMA OPERACIONAL

O monitoramento de erros pode ser muito útil na antecipação de falhas de hardware. Como, em plantas industriais, uma parada programada deve ser agendada para executar manutenção de hardware, é melhor que seja feita nessa janela do que às pressas no caso de uma falha de um hardware que esteja em produção.
Erros que podem indicar falha de hardware são erros de comprometimento/alocação de memória, leitura/escrita de disco, temperatura da CPU e velocidade da ventoinha, temperatura do disco, temperatura da memória, etc.

2.2 MONITORANDO PROCESSOS

Os processos-chave também podem ser monitorados a fim de verificar se eles não deixaram de funcionar para que a equipe seja alertada assim que ele falhe ou ainda, reiniciá-lo automaticamente (deve ser usado com extrema cautela, pois pode causar inconsistências).
Nomes e portas de processos suspeitos também podem ser monitorados, como RDP, HTTP/HTTPS, TeamViewer, "cmd.exe" e os processos do Windows PowerShell, e assim, desencadeando um alerta caso apresente sinais que poderiam indicar acesso remoto não autorizado.

2.3 MONITORANDO ALTA DISPONIBILIDADE

O estado dos links de comunicação pode ser verificado para ver se a rede da planta entrou em estado de contingência. O agente de monitoramento pode executar tarefas automatizadas, se necessário.

2.4 MONITORANDO O TRÁFEGO MODBUS

Você pode configurar um servidor para agir como um *sniffer* de rede, espelhando todo o tráfego de Modbus para a sua porta do switch. Um simples sniffer de Modbus pode ser construído usando python e scapy, a fim fazer *dump* de códigos de função, origens e destinos.

Com o monitoramento Modbus é possível criar alertas sobre os códigos de função não permitidos, os valores de tags e de origem e também pode ter a representação gráfica da frequência de comandos enviados e recebidos.

2.5 MONITORAMENTO DE TRAPS SNMP DOS CLP'S

Alguns CLPs oferecem monitoramento por SNMP (*Simple Network Management Protocol*). Itens como I/O de rede, pacotes descartados, protocolos desconhecidos recebidos, erros de rede, tabelas de alocação (útil contra ARP poisoning) e fragmentação.
Esses indicadores (especialmente os de erro) podem rapidamente dizer se algo está acontecendo.

2.6 MONITORAMENTO DE PROTOCOLO ICMP (PING) DOS CLP'S

Para CLPs que não suportam o monitoramento SNMP, pode-se usar um monitoramento simples através do comando *ping* para detectar a conectividade do dispositivo e também o tempo de resposta, fatores que poderiam indicar uma sobrecarga na controladora devido um ataque de negação de serviço (DoS).

2.7 MONITORAMENTO DISTRIBUÍDO

Monitoramento com Zabbix (software de código aberto) pode ser configurado com monitoramento distribuído. Isso significa que todas as plantas de automação podem ter sua própria estação de monitoramento reportando dados até a estação central. Isso pode ser muito útil, pois você pode ter estações de monitoramento distribuídas auto-reguladas reportando à principal estação de monitoramento do escritório da empresa.

2.8 FREQÜÊNCIA DE VERIFICAÇÃO

Dependendo da carga que a máquina executa, os itens podem ser configurados para serem verificados num intervalo definido. Os servidores com carga mais leve pode ter checagens mais curtas (a cada 15 ou 30 segundos) e servidores com carga maior podem ter um controle com intervalo maior (1 minuto ou superior) para preservar o poder computacional da máquina e a largura de banda.

2.9 ALERTAS

Além de monitorar e plotar os dados, alertas via e-mail, SMS ou Jabber (o nome original do protocolo extensível de mensagens e presença-XMPP, tecnologia aberta para mensagens instantâneas e presença) podem ser configurados para alertar a equipe de resposta. Com um pouco de esforço, os alertas podem gerar avisos sonoros ou qualquer outro tipo de método de alerta.

2.10 ESCALONAMENTOS

Os alertas também podem ser configurados para serem escalonados a outras pessoas, no caso de a equipe de resposta primária levar muito tempo para resolver o problema. Se um *trigger* permanece ativo depois de um tempo configurado, alertas de e-mail podem ser enviados automaticamente para a equipe de resposta do escritório ou mesmo para equipe de resposta do fabricante ou gradualmente subir a árvore de hierarquia até que o problema seja resolvido.

2.11 ALERTA DE ITEM DE GRUPO

Itens que compartilham a mesma categoria podem ser agrupados criando alertas mais objetivos. É possível ter todos os alertas de relacionados a bancos de dados disparados para a equipe de banco de dados, os alarmes relacionados ao SCADA, disparados paraa equipe de automação e assim por diante. O escalonamento pode ser ampliado aqui também, assim os níveis mais elevados de suporte podem ser contatados caso a primeira equipe responsável não tenha conseguido resolvê-los a tempo.

3. A PLATAFORMA DE TESTES

Plataforma de testes (do inglês *testbed*) é o nome dado a uma plataforma de ensaio para a execução de experiências estruturadas de maneira segura e controlada. A estrutura utilizada para este trabalho é composta de elementos que simulam o comportamento de uma rede de automação réplica do mundo real dos processos industriais. Devido a fatores específicos para ambientes SCADA, como a criticidade de sistemas de tempo real e a necessidade de disponibilidade ininterrupta, plataformas de teste representam plataformas ideais para observar o comportamento de sistemas e analisar os componentes de sistemas de controle.

3.1 O AMBIENTE DE TESTES

A estrutura de testes existente no Laboratório TI Safe inclui o equipamento de campo - que consiste em um CLP Wago 741-800 e um hardware simulando uma planta industrial de gás natural (*Tofino Scada Security Simulator*), uma estação Windows 7 (física) agindo como a estação de supervisão , um servidor de monitoramento (Debian Linux 6) e um servidor de *sniffer* de tráfego Modbus (Debian Linux 6 com um script em python + scapy), as duas últimas em máquinas virtuais.

Figura 1: O Simulador de Segurança SCADA utilizado para os testes

3.1.1 A REDE DE TESTES

A rede configurada não tem segmentação nem por sub-redes, roteamento ou VLANs. Todos os equipamentos são conectados na mesma rede "flat" dentro da mesma faixa de endereços IP (192.168.1.0/24).

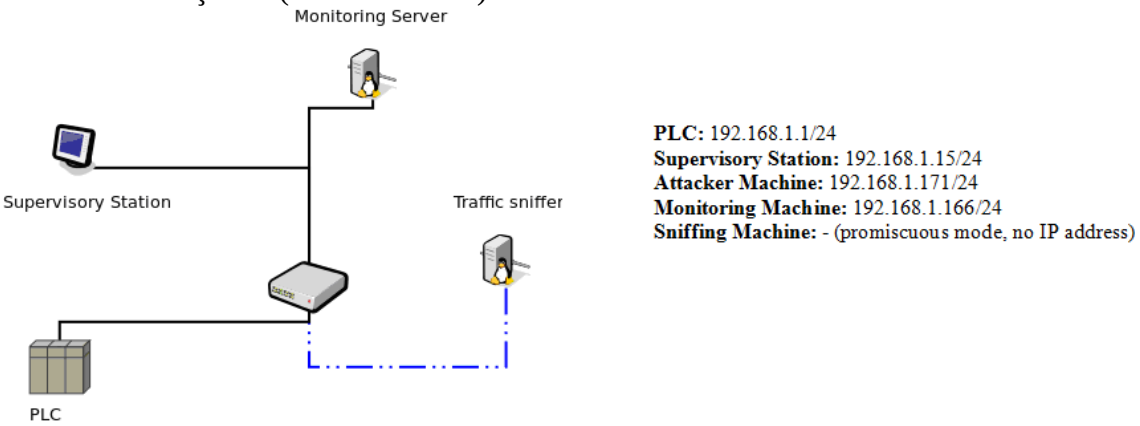


Diagrama 1: A rede de testes

3.1.2 A MÁQUINA ATACANTE

A máquina atacante é um laptop HP que será conectado diretamente ao switch (não mostrado no diagrama acima) e está rodando Kali Linux 1.0 a partir de um Live-CD. Abaixo está a lista dos softwares utilizados nos testes:

Software / Ferramenta	Descrição	Ataque	Autor
Hping3	Ferramenta de ICMP flood	Negação de serviço em camada 3	http://www.hping.org/
T50	Ferramenta de flood	Negação de serviço em camada 3	https://github.com/merces/t50
Meterpreter	Shell de acesso remoto	Comprometimento remoto, infecção por malware	http://www.metasploit.com/
Arpspoof	Ferramenta de ARP poison/spoofing	ARP poison	http://arpspoof.sourceforge.net/
Pymodbus	Biblioteca python para Modbus	Tráfego Modbus não autorizado	https://github.com/bashwork/pymodbus

Tabela 2: Lista de softwares usados nos testes

3.1.3 O SERVIDOR DE MONITORAMENTO

O servidor de monitoramento é baseado em uma máquina Debian Linux rodando a solução open source de monitoramento Zabbix 2.0.6 e MySQL 5.1 como backend de dados. Idealmente em um ambiente de produção, os backends das soluções de

monitoramento seriam divididas entre vários servidores por motivos de desempenho e isolamento.

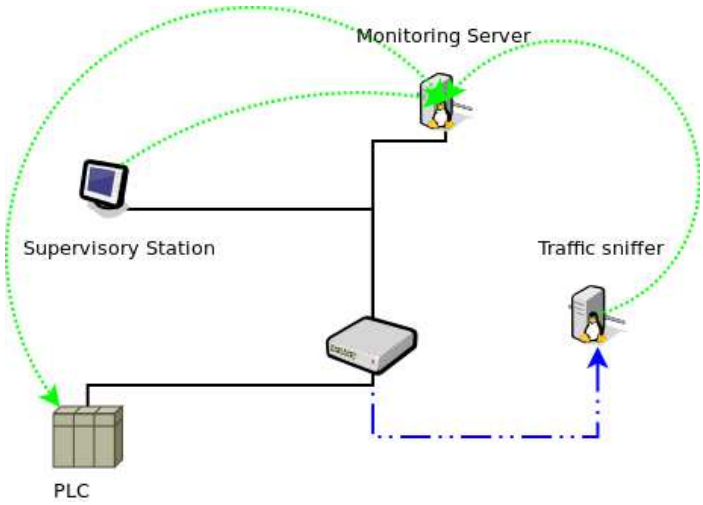


Diagrama 2: Fluxo de dados de monitoramento. A seta indica se os dados são coletados remotamente ou enviados pelo agente do ativo.

Os dados são coletados ativamente ou por agentes Zabbix ou passivamente por consultas ICMP e SNMP. Os dados recolhidos podem depois ser alimentados diretamente para o Zabbix onde serão normalizados, representados graficamente e armazenados.

As MIBs (*Management Information Bases*) SNMP no CLP foram enumeradas com a ferramenta snmpwalk e foram convertidas para o template Zabbix com o script perl zload_snmpwalk (https://www.zabbix.com/wiki/howto/monitor/snmp/zload_snmpwalk).

3.1.4 O SNIFFER DE REDE

O sniffer de rede consiste em uma instalação Linux simples ligado a uma porta específica do switch que está configurada para receber o espelhamento de qualquer outra porta em que exista tráfego Modbus. Um sniffer programado internamente em python dentro do scapy (http://www.secdev.org/projects/scapy/), programa de manipulação de pacotes. O sniffer é capaz de decodificar o tráfego Modbus e exibir de informações no seguinte formato:

{'request': '0330000064', 'unit_id': 1, 'src_ip': '192.168.1.15',
'dst_ip': '192.168.1.1', 'response':
'03c800060050001600120190001200500050001600120012019011 30cd00cd00cd00cd00
cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00c
d00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd00cd
00cd00cd00cd0088010040081100000120104800008000000240000002000000002000000
0000020041000045020000000020000002000400200000000200040800000040800200000
0020000000000800cd0010800000cd0000440000', 'func_code': 3}

O sniffer é composto por três módulos, disparados como instâncias bifurcadas neste paradigma multiprocessado: o "Sniffer", os "Workers" e o "Publisher". Estes foram construídos desta maneira para deixar que as operações sejam executadas isoladamente e sem um bloquear o outro.

Os principais processos são o sniffer construído em cima de scapy. Este módulo alimenta uma fila IPC que é posteriormente consumida pelos Workers (100 instancias) que irá adicionar as transações ao pool para a sumarização. O Publisher recebe o resumo, relata para o Zabbix para monitoramento e geração de gráficos e, em seguida, descarta tudo, e assim o ciclo começa novamente.

Desta forma, os alertas podem ser configurados se forem detectados códigos de função inválidos. O sniffer pode ser também reprogramado para transmitir mais dados, como valores de registro de escrita e outros.

3.2 ATAQUES REALIZADOS

Os seguintes ataques foram realizados contra os ativos de rede industrial simulada:

Ataque	Vetor de ataque	Ativos afetados
Interceptação de comunicações	ARP poison	CLP, Estações Supervisórias
Negação de Serviço em CLP	Flood em camada 3, 0day	CLP
Infecção por malware na Estação Supervisória	Malware para Modbus e Meterpreter shell backdoor	Estações Supervisórias, Rede
Comprometimento de Estação Supervisória	Meterpreter shell backdoor	Estações Supervisórias
Logon remoto não autorizado	Habilitando o remote desktop na máquina, acessando uma máquina a partir de outra máquina na rede	Estação Supervisória
Tráfego Modbus não autorizado	O envio de comandos a partir da máquina atacante	CLP

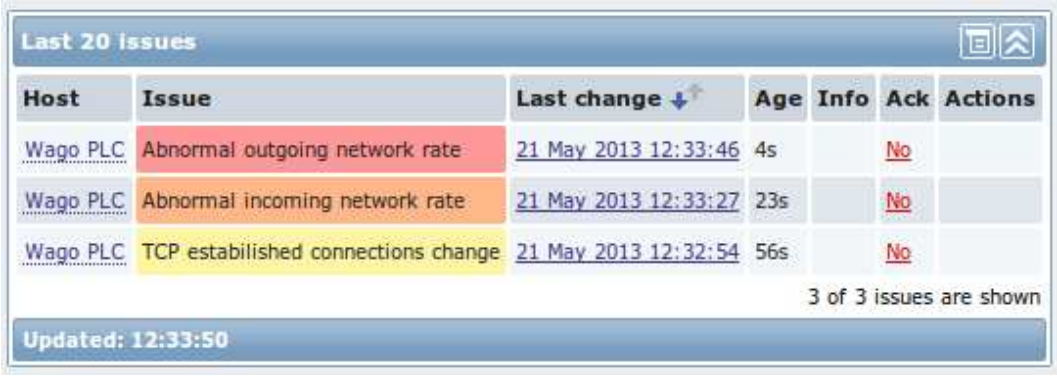
Tabela 3: Ataques realizados.

3.3 RESULTADOS DOS TESTES

Como testar geralmente envolve alguma coleta de informação, começamos os testes através da emissão de algumas varredura simples com o *nmap*.

```
$ nmap -sV 192.168.1.1
```

Sem qualquer limitação de taxa, três triggers foram ativadas: tráfego de entrada anormal, tráfego de saída anormal e mudança de número da conexão TCP.



Last 20 issues

Host	Issue	Last change	Age	Info	Ack	Actions
Wago PLC	Abnormal outgoing network rate	21 May 2013 12:33:46	4s		No	
Wago PLC	Abnormal incoming network rate	21 May 2013 12:33:27	23s		No	
Wago PLC	TCP estabilished connections change	21 May 2013 12:32:54	56s		No	

3 of 3 issues are shown

Updated: 12:33:50

Imagem 2: *Triggers ativadas por varredura nmap*

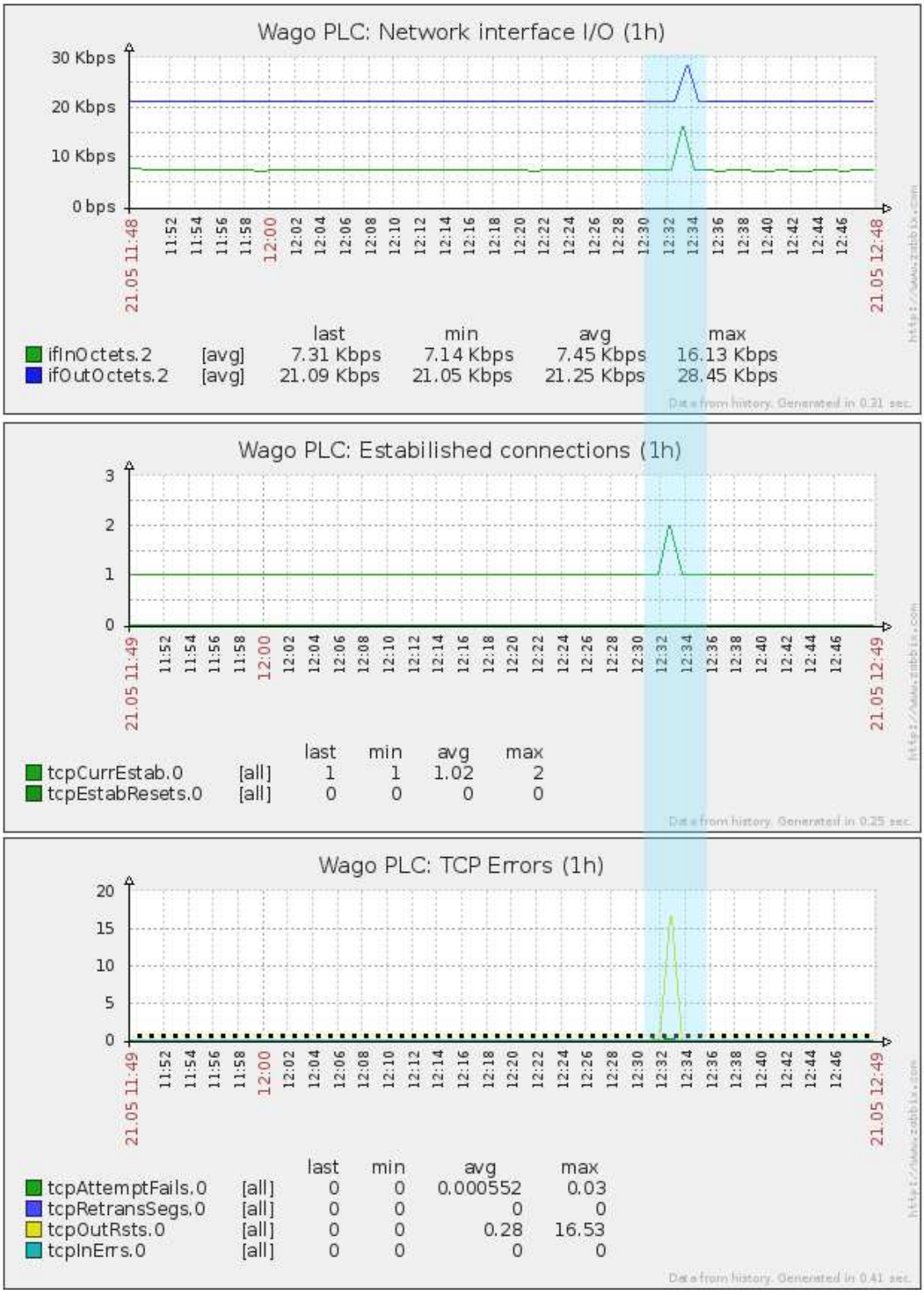


Imagem 3: Picos gerados pela varredura nmap em TX/RX e alguns TCP RSTs liberados.

3.3.1 INTERCEPTAÇÃO DE COMUNICAÇÃO

Nosso script de verificação de tabela ARP fez o seu trabalho e informou o endereço MAC trocado em 192.168.1.1 (CLP) quando a estação de supervisão foi envenenada usando *arpspoof*.

Figura 4: *Trigger para mudanças ARP*

A entrada UserParameter a seguir foi adicionada ao arquivo de configuração do agente Zabbix da estação de supervisão, para o teste de ARP:

```
UserParameter = arpcheck [*], ping-n 1-w 1 $ 1> NUL e for / f ″tokens = 2″% i in ('arp-a | findstr / r ″$ 1 ¥>″') do @ echo% i
```

Assim, os itens podem ser criados como "arpcheck [192.168.1.1]" para obter o endereço MAC na tabela ARP para este IP. Um *trigger* é criado para disparar quando houver alteração deste valor.

3.3.2 NEGAÇÃO DE SERVIÇO EM UM CLP

O ataque ICMP flood foi muito ruidoso, como o esperado. Várias *triggers* foram disparadas durante o ataque. Os gráficos mostram claramente picos anormais.

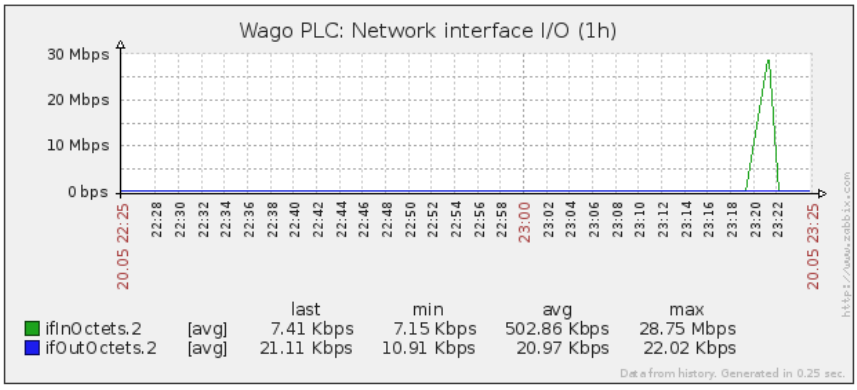


Figura 5: Pico verde mostra 30Mbps de pico devido ao flood

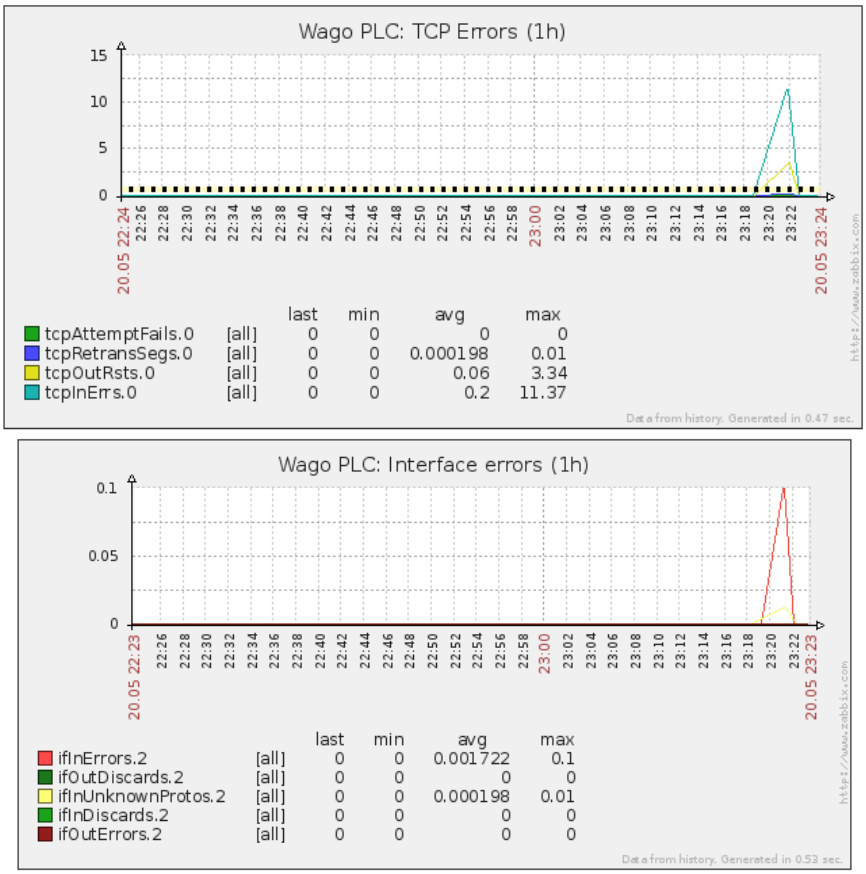


Figura 6: Erros causados pela sobrecarga na rede

Os dados SNMP não ficam disponíveis enquanto o CLP está sob ataque DoS pois o cliente SNMP não pode se conectar ao dispositivo de coleta de dados e então essas triggers vão desaparecer após o ataque ter cessado e são consideradas triggers colaterais.

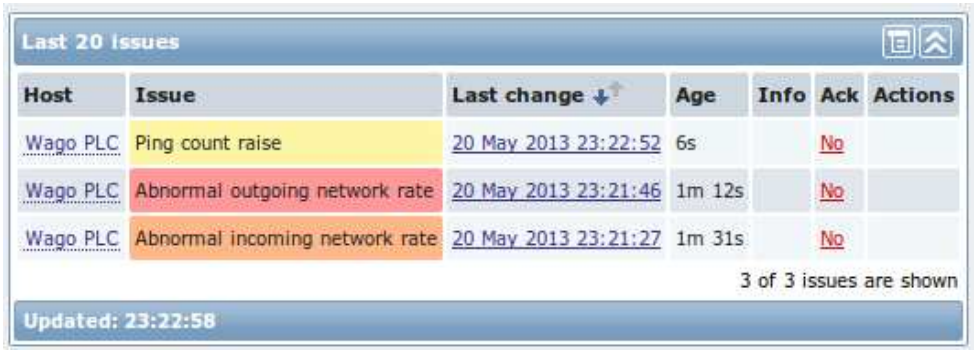


Figura 7: Triggers emitidas após o ataque DoS

Para ser alertado assim que o dispositivo começar a ser atacado por um ataque Denial-of-Service, nós periodicamente disparamos um *ping* para o CLP. Se o CLP não responder dentro de um certo *timeout*, ele é considerado desligado e uma trigger é emitida.

Figura 8: Trigger de CLP com falha no ping

3.3.3 INFECÇÃO POR MALWARE NA ESTAÇÃO DE SUPERVISÃO

Como o tráfego de rede de automação é bem homogêneo, podemos definir limites mais estreitos na entrada de rede e variância de saída. A maioria das ferramentas de acesso remoto (RATs) não se importam em limitar a sua velocidade e provavelmente vão tentar se comunicar o mais rápido que puderem.

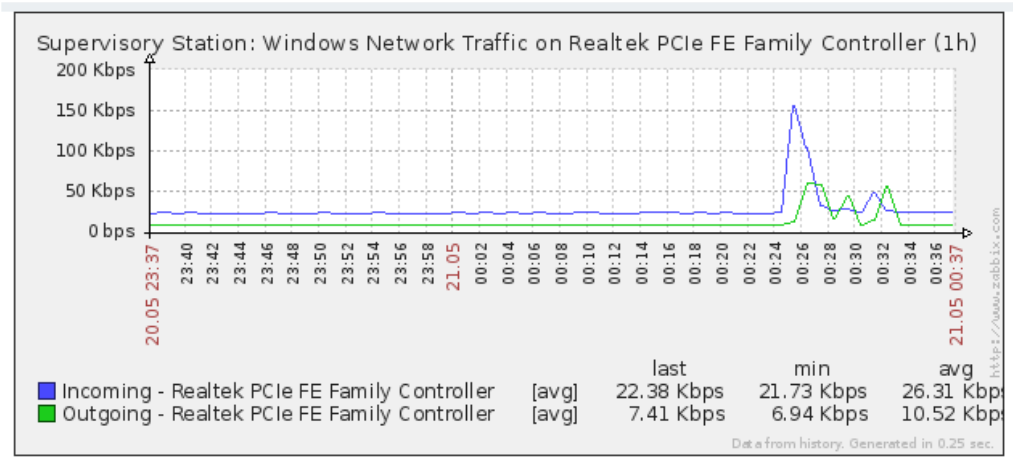


Figura 9: O tráfego de rede salta em uma sessão de Meterpreter

Figura 10: Triggers de tráfego de rede anormal

A sessão Meterpreter gera um pouco de ruído enquanto baixa o estágio, mas quando se começa a enviar alguns comandos (como o 'ls', 'ps', 'migrate' e alguns scripts) o gráfico fica muito fora dessa linha quase plana, mostrando que alguém está fazendo alguma coisa na estação de supervisão.

Entrada e saída de rede são os melhores lugares para detectar precocemente surtos de malware já que worms são geralmente barulhentos pois tentam se comunicar com seu emissor ou se propagar através da rede.

3.3.4 COMPROMETIMENTO DE UMA ESTAÇÃO DE SUPERVISÃO

Criamos um item de monitoramento para o número de processos "cmd.exe" em execução a fim de ser notificado a cada 30 segundos se um cmd.exe está aberto em qualquer estação de supervisão. Normalmente nenhum *shell* deve ser aberto a menos que o sistema esteja sob algum tipo de manutenção.

Figura 11: Trigger de notificação sobre um novo shell aberto

O Windows PowerShell também é monitorado uma vez que é menos comum de ser usado regularmente nas estações de supervisão.

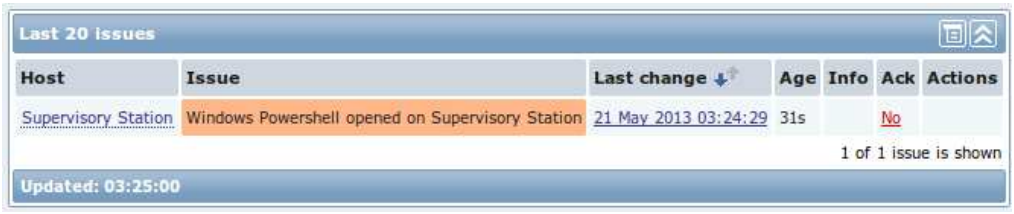


Figura 12: Trigger notifica sobre novo Windows PowerShell aberto

Quando o sistema é marcado como "em manutenção", as triggers são suprimidas, assim é possível abrir quantas *shells* forem necessárias.

Se for necessário executar tarefas programadas em lotes, você pode adicionar intervalos de tempo onde é permitido ter um ou mais processos "cmd.exe" em execução.

3.3.5 ACESSO REMOTO NÃO AUTORIZADO

Ao monitorar os logs de eventos do Windows, podemos determinar se uma nova sessão é criada. Como podemos notar, nossa trigger detectou o novo acesso imediatamente.

Figura 13: Trigger para novas sessões criadas na estação Windows

3.3.6 TRÁFEGO MODBUS NÃO AUTORIZADO

O tráfego não autorizado causou mudanças sutis, mas perceptíveis no gráfico TX / RX. Se a rede de automação mantiver um tráfego constante, limites de variação podem ser ajustados para detectar tráfegos anômalos.

O pico dentro da área marcada em azul é onde os comandos não autorizados foram enviados. Observe o aumento da contagem de conexões TCP e de tráfego durante este período.

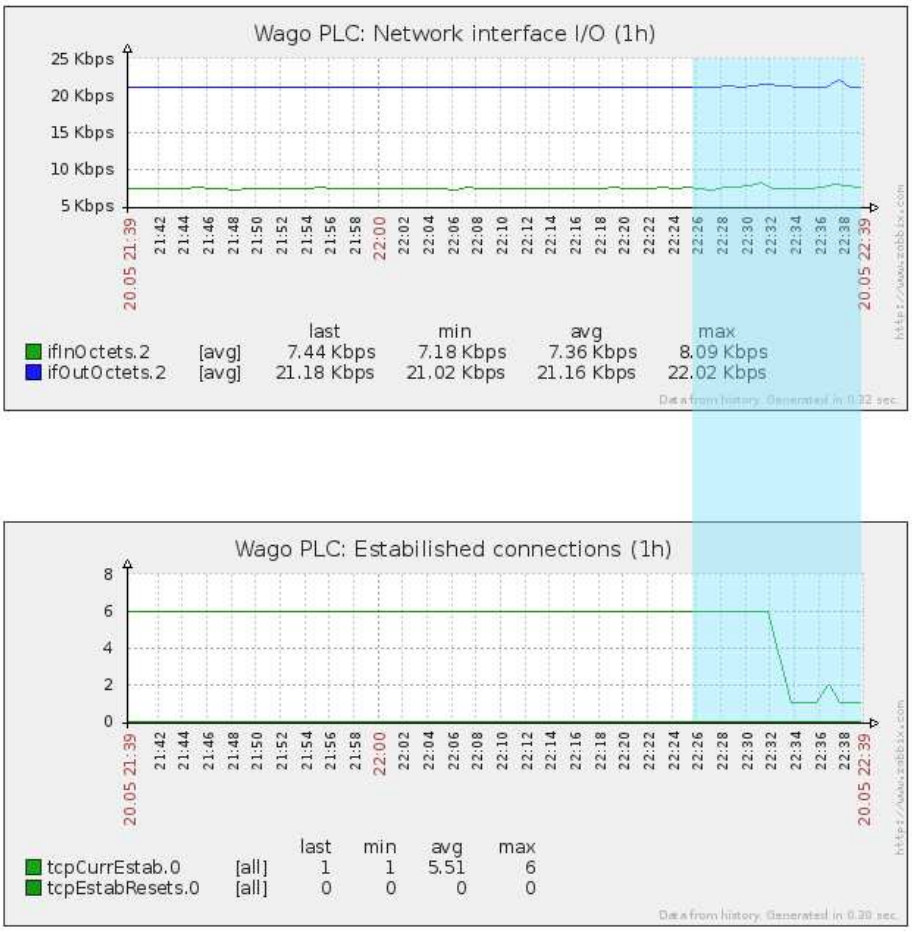


Figura 14: *Picos gerados pelo tráfego Modbus da máquina atacante*

Figura 15: Triggers disparadas pelo tráfego anormal gerado pela emissão de tráfego modbus não autorizado e extração de dados modbus

O *sniffer* de rede também nos dá uma boa visualização de códigos de função Modbus. Veja o código de função 3 (Ler Múltiplos Registros), por exemplo. A Estação de supervisão solicita dados a cada *N* segundos para atualizar o software de supervisão. O gráfico é muito constante, como mostrado abaixo.

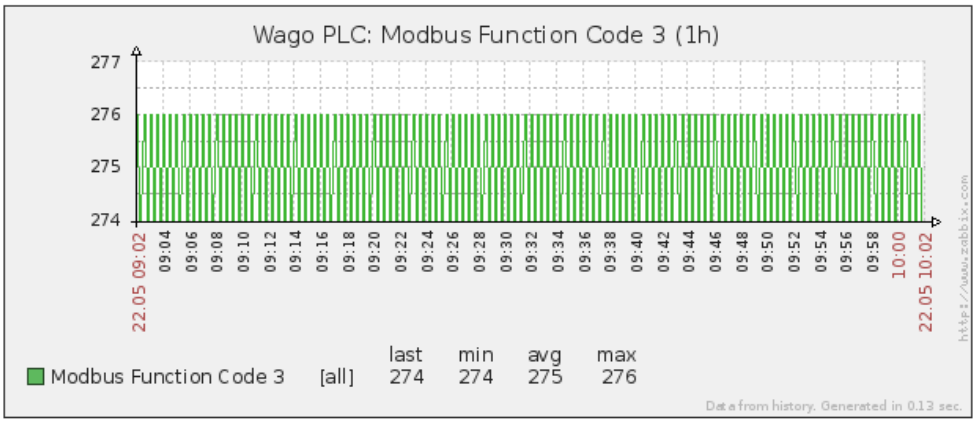


Figura 16: Código regular de função 3 Modbus tráfego para CLP

Assim que o atacante começa a enviar funções Modbus de código ao CLP, a fim de enumerar *tags*, o gráfico cria picos (destaque abaixo) que avisam sobre nossa enumeração.

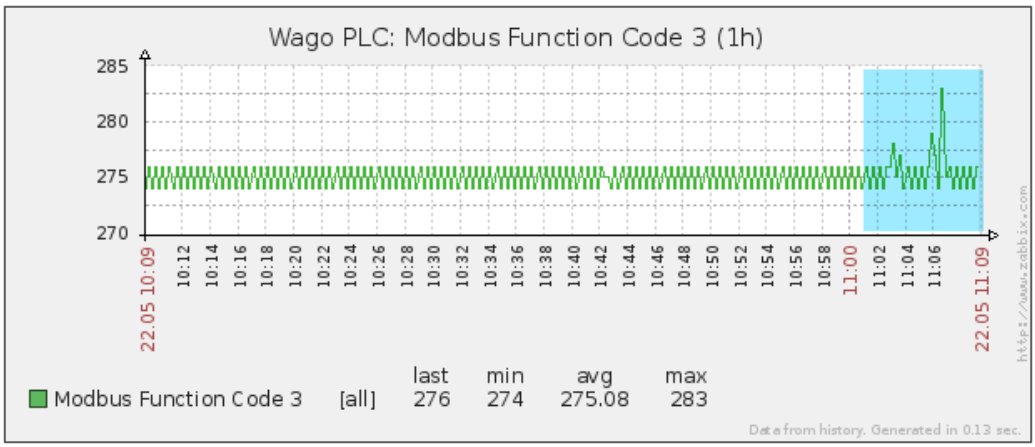


Figura 17: Picos após sondagem Modbus

O primeiro pico é devido a sondagem manual e individual de *tags* via linha de comando do cliente Modbus. O segundo (maior) é devido ao script "enum.sh" que tenta ler tags de um intervalo fornecido. Como a comunicação normal é constante, essa mudança sutil também aciona uma trigger.

Figura 18: Trigger acionada pelo tráfego Modbus não autorizado

4. CONCLUSÃO

A homogeneidade do comportamento cíclico de redes e servidores industriais permite-nos estabelecer com pouco esforço os parâmetros 'saudáveis' da rede. Esta mesma característica dificilmente é encontrada em redes de TI por sua natureza de uso e torna inviável realizar um monitoramento com o mesmo nível de precisão sem a enorme ocorrência de falsos-positivos.

Aplicações de análise e monitoramento de rede e servidores são fundamentais para a detecção de tráfegos de rede incomuns, realizando a administração da rede de sistemas de controle, e auxiliando na resposta a incidentes de segurança. Este tipo de software endereça uma necessidade geral de segurança em sistemas de controle ao invés de vulnerabilidades específicas.

Ao realizar monitoramento por comportamento consegue-se obter resultados mais tangíveis do que o monitoramento por *keywords* conhecidas, as chamadas 'assinaturas'.

Quando aplicado em uma rede de automação, o software de monitoramento pode ser usado para estabelecer uma linha de base de tráfego de rede normal, uma tarefa que ajuda a facilitar a resposta a incidentes e avaliação de riscos. O estabelecimento de linhas de base de tráfego por meio de análise de pacotes na rede de sistemas de controle é necessário para a detecção de tráfego anômalo através da análise das diferenças.

Uma vez que o tráfego de rede irregular tenha sido capturado e analisado pelo software de monitoramento, a equipe de segurança usará os *dumps* de dados para avaliar o que realmente está acontecendo na rede. O Tráfego anômalo será comparado ao tráfego da linha de base para fornecer informações importantes sobre quais servidores (ou equipamentos) estão gerando o tráfego anômalo, que portas e serviços podem estar envolvidos, e quais protocolos de rede estão sendo usados. *Dumps* de pacotes de tráfego incomum podem ser usados para determinar se o tráfego é devido a problemas de rede, erros de configuração do sistema, ou um sistema comprometido.

Depois que a atividade normal da rede é demarcada, gatilhos (*triggers*) podem ser configurados para parâmetros fora destas faixas que podem significar um comprometimento dos ativos em questão. Baseados em triggers, alarmes (inclusive sonoros) podem ser configurados.

Para ambientes de automação industrial e controle, com seus protocolos não-usuais, há poucas ferramentas comerciais disponíveis para a compra, e o ideal é customizar uma ferramenta de código aberto para sua própria necessidade de monitoramento.

REFERÊNCIAS NA INTERNET

1. http://www.tisafe.com/en/solucoes/governanca-industrial/
2. http://www.tofinosecurity.com/
3. https://www.zabbix.org
4. https://www.zabbix.com/documentation/2.0/manual/config/items/itemtypes/zabbix_agent/win_keys
5. https://www.zabbix.com/forum/showthread.php?t=10679
6. https://www.zabbix.com/wiki/howto/monitor/snmp/zload_snmpwalk
7. http://technet.microsoft.com/en-us/library/dd941635 (v = WS.10). aspx
8. http://technet.microsoft.com/en-us/library/cc732459 (v = WS.10). aspx
9. http://www.kali.org
10. http://docs.python.org/2/library/multiprocessing.html
11. https://www.zabbix.com/forum/showthread.php?p=90132