Authenticated Network Time Synchronization

Benjamin Dowling, Douglas Stebila, Greg Zaverucha

Acadêmico: Weverton Bueno da Silva

Network Time Protocol (NTP)

- Um dos protocolos mais antigos (RFC 958 - 1985);
- Permite a sincronização dos relógios dos dispositivos de uma rede à partir de referências de tempo confiáveis;

Funcionamento

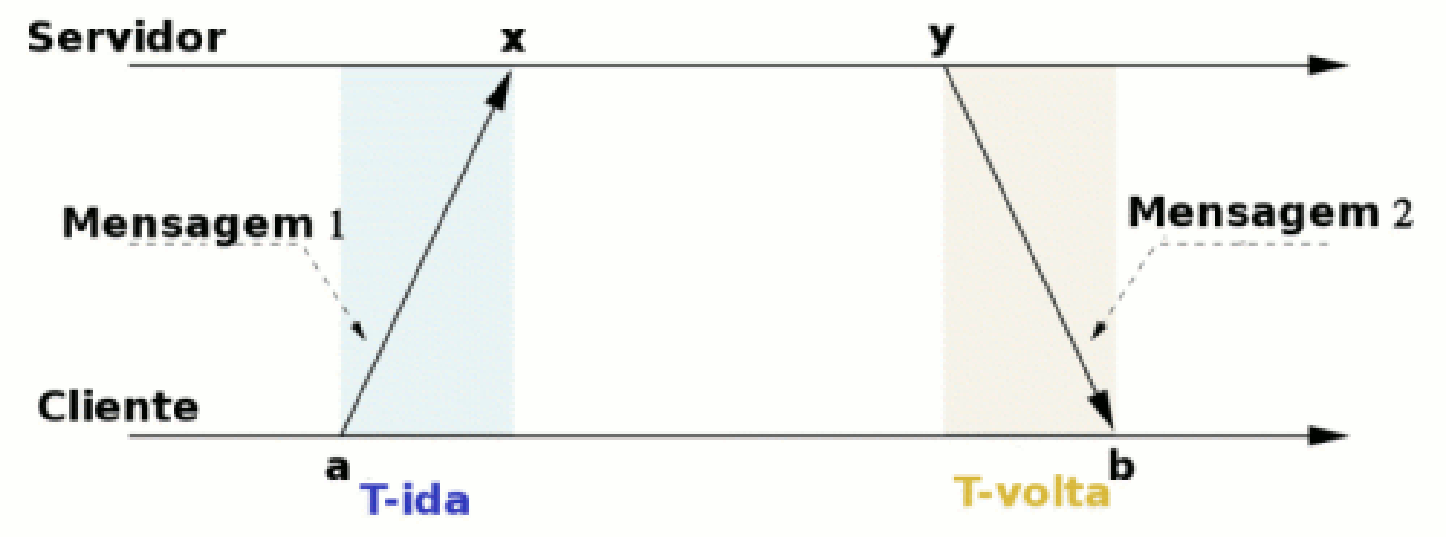

Figura 1: Troca de Mensagens

Calculo do Atraso/Offset

- Atraso:

$$Atraso = (b - a) - (y - x)$$

- Offset (Considerando que T-ida = T-volta):

$$Offset = x - \left(a + \frac{atraso}{2}\right)$$

$$Offset = \frac{x - a + y - b}{2}$$

Importância na Segurança

- Validação de certificados em TLS e outros protocolos;
- Verificação de *tickets* do protocolo *Kerberos;*
- HTTP Strict Transport Security (HSTS).

Segurança Atual

- Versões antigas (NTPv0-NTPv2) não possuem métodos de autenticação;
- Na versão NTPv3 foi adicionado um método de autenticação usando uma chave de criptografia simétrica pré-compartilhada;
- No NTPv4 foi introduzido o Autokey, um método de autenticação por chaves públicas;
- Recentemente foi proposto o Network Time Security (NTS).

Contribuição

- O protocolo ANTP permite que o servidor se autentique a um cliente usando certificados de chaves publicas e troca de chaves publicas, além de proporcionar uma garantia de que o pacote não foi alterado no transporte;
- É usado um método fora da comunicação para verificar se o certificado é valido;
- Segue a RFC 7384.

Funcionamento

- Possui três fases:

- Negociação: Cliente e Servidor decidem qual algoritmo de criptografia usar.
- Troca de chaves: Cliente e Servidor usam chaves publicas para criar uma chave simétrica.
- Sincronização do tempo: Cliente envia uma requisição e o servidor envia um pacote autenticado com a chave simétrica obtida anteriormente.

- Pode ser utilizado em modo *"no-cryptographic-latency"*, onde o servidor envia dois pacotes de resposta, sendo que o primeiro não possui autenticação e o segundo pacote possui.

- A fase de sincronização do ANTP é quase tão rápida quanto a requisição NTP simples não autenticada (cerca de 21 µs mais lenta com uma carga de 50% do servidor).

Client		**Server**
supported algorithms $\vec{alg}_C$		supported algorithms $\vec{alg}_S$ long-term secret s certificate $cert_S$ for the KEM keypair (pk_S, sk_S)
	Negotiation phase	
$\alpha \leftarrow$ `in-progress` $n_c \leftarrow\!\!\$ \{0,1\}^{256}$ $m_1 \leftarrow \vec{alg}_C \| n_c$	$\xrightarrow{m_1}$	$(\mathsf{KDF}, \mathsf{Hash}, \mathsf{KEM}, \mathsf{MAC}) \leftarrow \text{negotiate}(\vec{alg}_C, \vec{alg}_S)$ $h \leftarrow \mathsf{Hash}(m_1 \| \vec{alg}_S \| cert_S)$ $C_1 \leftarrow \mathsf{AuthEnc}_s(01 \| h \| \mathsf{KDF} \| \mathsf{Hash} \| \mathsf{KEM} \| \mathsf{MAC})$
Verify $cert_S$ $pk_S \leftarrow \text{parse}(cert)$	$\xleftarrow{m_2}$	$m_2 \leftarrow \vec{alg}_S \| cert_S \| C_1$

Key exchange phase

$(\mathsf{KDF}, \mathsf{Hash}, \mathsf{KEM}, \mathsf{MAC}) \leftarrow \text{negotiate}(\vec{algs}_C, \vec{algs}_S)$		
$h \leftarrow \mathsf{Hash}(m_1 \Vert \vec{algs}_S \Vert cert_S)$		
$(e, pms) \leftarrow \mathsf{KEM.Encap}(pk_s)$		
$m_3 \leftarrow C_1 \Vert e$	$\xrightarrow{m_3}$	$b \Vert h \Vert \mathsf{KDF} \Vert \mathsf{Hash} \Vert \mathsf{KEM} \Vert \mathsf{MAC} \leftarrow \mathsf{AuthDec}_s(C_1)$
		If $b \neq 01$, then $\alpha \leftarrow$ `reject` and abort
		$pms \leftarrow \mathsf{KEM.Decap}(sk_S, e)$
$k \leftarrow \mathsf{KDF}(pms, \perp, \text{“ANTP”}, len)$		$k \leftarrow \mathsf{KDF}(pms, \perp, \text{“ANTP”}, len)$
		$C_2 \leftarrow \mathsf{AuthEnc}_s(02 \Vert k \Vert \mathsf{KDF} \Vert \mathsf{Hash} \Vert \mathsf{KEM} \Vert \mathsf{MAC})$
		$\tau_1 \leftarrow \mathsf{MAC}(k, h \Vert m_3 \Vert C_2)$
Verify $\tau_1 = \mathsf{MAC}(k, h \Vert m_3 \Vert C_2)$	$\xleftarrow{m_4}$	$m_4 \leftarrow C_2 \Vert \tau_1$
If verify fails, then $\alpha \leftarrow$ `reject` and abort		

Time synchronization phase $p = 1, \ldots, n$

$\alpha \leftarrow$ `in-progress`		
$n_{c_2} \xleftarrow{s} \{0,1\}^{256}$		
$t_1 \leftarrow \mathsf{Now}()$		
$m_5 \leftarrow t_1 \Vert n_{c_2} \Vert C_2$	$\xrightarrow{m_5}$	$t_2 \leftarrow \mathsf{Now}()$
		$b \Vert k \Vert \mathsf{KDF} \Vert \mathsf{Hash} \Vert \mathsf{KEM} \Vert \mathsf{MAC} \leftarrow \mathsf{AuthDec}_s(s, C_2)$
		If $b \neq 02$, then $\alpha \leftarrow$ `reject` or abort
		$t_3 \leftarrow \mathsf{Now}()$
	$\left[\xleftarrow{m_6^*}\right]$	$m_6^* \leftarrow t_1 \Vert t_2 \Vert t_3$
		$\tau_2 \leftarrow \mathsf{MAC}(k, m_5 \Vert t_1 \Vert t_2 \Vert t_3)$
$t_4 \leftarrow \mathsf{Now}()$	$\xleftarrow{m_6}$	$m_6 \leftarrow t_1 \Vert t_2 \Vert t_3 \Vert \tau_2$
$RTT \leftarrow (t_4 - t_1) - (t_3 - t_2)$		
If $RTT > E$, then $\alpha \leftarrow$ `reject` and abort		
Verify $\tau_2 = \mathsf{MAC}(k, m_5 \Vert t_1 \Vert t_2 \Vert t_3)$		
If verify fails, then $\alpha \leftarrow$ `reject` and abort		
$offset = \frac{1}{2}(t_3 + t_2 - t_1 - t_4)$		
$time_p \leftarrow \mathsf{Now}() + offset$		
$\alpha \leftarrow \mathtt{accept}_p$		
If $p = n$, then terminate		

Vantagens

- O cliente pode autenticar o servidor e todas as mensagens enviadas, evitando ataques de retransmissão (Replay attacks);
- O servidor não precisa manter o estado de cada cliente;
- É realizada apenas uma operação de chave pública por cliente;
- A chave secreta pode ser utilizada para várias sincronizações do mesmo cliente;
- O cliente possui a opção “sem criptografia” para evitar o erro causado pelas operações de criptografia.

Implementação

- OpenNTPD v1.92;
- OpenSSL v1.0.2f – biblioteca libcrypto;

- Criptografia simétrica: AES128-GCM;
- Hash: SHA-256;
- MAC: HMACSHA256;
- Enclapsulamento: RSA key transport ou Static-ephemeral elliptic curve Diffie–Hellman.

Testes

LAN:

- Duas máquinas cliente;
- Um servidor com Linux Mint 17.2;
- Conexão de 1 Gb/s.

Longa distância:

- Um cliente e um servidor (ambos da Amazon) usando Ubuntu 14.04;
- Cliente na Virginia (US East) e servidor na California (US West).

Resultados

Phase	Throughput	Latency within LAN (μs)		Latency across US (ms)	
		50% load	90% load	50% load	90% load
ANTP – Negotiation – RSA	58 240	186 ± 26	202 ± 44	76.3 ± 0.1	77.5 ± 0.1
ANTP – Negotiation – ECDH	146 808	172 ± 35	233 ± 133	75.3 ± 0.1	75.3 ± 0.1
ANTP – Key Exchange – RSA	1 754	891 ± 125	997 ± 348	75.8 ± 0.2	76.9 ± 0.5
ANTP – Key Exchange – ECDH	13 210	197 ± 56	344 ± 142	74.7 ± 0.2	75.4 ± 0.4
ANTP – Time Synchronization	175 644	168 ± 35	230 ± 160	73.5 ± 0.1	73.7 ± 0.1
ANTP – All 3 phases – RSA	–	2255 ± 587	2646 ± 345	226.6 ± 6.2	258.0 ± 35
ANTP – All 3 phases – ECDH	–	1325 ± 499	2252 ± 1172	231.8 ± 10.5	223.3 ± 6.7
NTP	291 926	147 ± 34	181 ± 136	72.4 ± 0.1	74.0 ± 0.1

Referências

- Benjamin Dowling, Douglas Stebila, Greg Zaverucha. Authenticated Network Time Synchronization. Disponível em:

  https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_dowling.pdf.

- http://ntp.br/ntp.php