Skype

Analýza funkcionality a zabezpečení

Fakulta informatiky
Masarykova univerzita

29. dubna 2010

Členové týmu

- Petr Bartel
- Tomáš Král
- Ladislav Tkáč
- Jiří Vomáčka
- Tomáš Pyszko
- Marek Čermák
- Richard Nossek

Osnova

Úvod

- Historie.
- VoIP, P2P.
- Architektura sítě.
- Detekce provozu, IDS, FW.

Historický úvod

- zakladatelé Janus Friis a Niklas Zennstrom, rok 2003
- Skype technologies S.A. je registrováno v Lucembursku
- close-source, VoIP aplikace (postavená na principech P2P)
- multiplatformní – (Linux, MacOS, iPhone OS, Maemo, Windows Mobile, Symbian, Android, Playstation, Windows)
- Ve Q3 2009 27,7 milionů minut hovorů, 521 milionů účtů, s více než 20 miliony aktivních uživatelů za den.
- 8% všech světových hovorů

Internetová telefonie v podání Skype

- VoIP hovory – hlavní funkcionalita, P2P infrastruktura
- videokonference
- instant messaging
- hovory do běžných telefonních sítí
- přenos souborů
- hry a doplňky
- proprietární bez dostupné dokumentace

Architektura sítě

- Node
- Supernode
- Login server
- Host cache

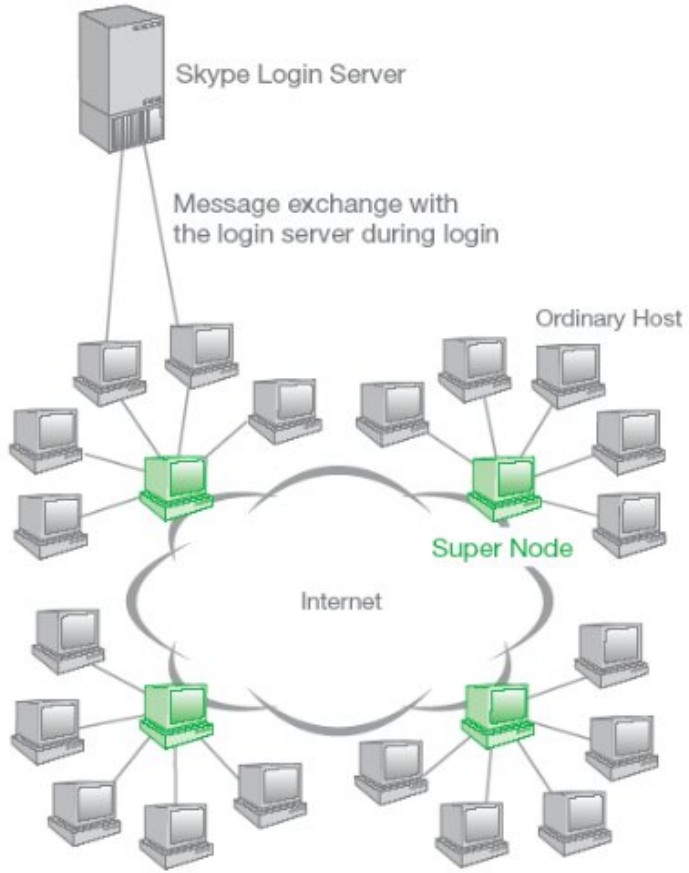

Figure 1 - Skype Network: Ordinary Hosts and Supernodes

Soukromí se Skype

- Kurt Sauer “We provide a safe communication option, I will not tell you whether we can listen or not.“
- Speciální verze v Číně (TOM) - filtruje instant messaging
- Síťový provoz skypu nelze jednoduše izolovat
- Dokáže obejít firewally, NAT i proxy.
- Dokonalý backdoor, díky využití kryptografie.
- V korporátním prostředí – připojení k vnější entitě, často proti bezpečnostní politice

Detekce provozu Skype

- Existují pouze proprietární produkty pro firemní detekci Skypu vzhledem ke složitosti protokolu a snaze znemožnit reverzování.
- Veřejná alternativa je založena na Pearsonovu Chi-kvadrát testu s přihlédnutím k charakteristikám VoIP trafficu. Umožňuje vytvořit jakýsi otisk charakteristiky paketů a obejít tak náhodnost na bitové úrovni vzniklou šifrováním.
- Jinou možností je stochastická charakterizace v závislosti na délce a počtu příchozích paketů. Toho se poté využije v rozhodovacím procesu založeném na naivním Bayesiánském klasifikátoru.

Blokace Skype

- inspekce TCP i UDP payloadu paketů pomocí např. nástroje Snort
- pakety s hodnotou 0x1703010000 jsou odpovědné za navázaní spojeni
- nemá vliv na samotnou komunikaci pomocí Skype

Registrace

- Klient:
  1. uživatel si zvolí unikátní uživatelské jméno a heslo
  2. zašle uživatelské jméno a SHA-1 hash hesla na Skype Login Server (zašifrováno veřejným klíčem serveru)
- Skype login server:
  1. pomocí privátního klíče dešifruje zprávu a získá uživatelské jméno a hash hesla

Registrace

- Klient:
  1. uživatel si zvolí unikátní uživatelské jméno a heslo
  2. zašle uživatelské jméno a SHA-1 hash hesla na Skype Login Server (zašifrováno veřejným klíčem serveru)
- Skype login server:
  1. pomocí privátního klíče dešifruje zprávu a získá uživatelské jméno a hash hesla

Registrační proces

- Centrální server disponuje veřejným a privátním klíčem (S_s a V_s)
- Veřejný klíč je zabudován do klienta
- Uživatel A vygeneruje privátní a veřejný klíč (S_a, V_a)
- Rovněž vygeneruje hash $H(Pa)$ - který si ponechá lokálně uložený a pošle přes AES kanál serveru S tři údaje: A, $H(Pa), V_a$
- S uloží pár (pokud je jedinečný) A, $H(H(Pa))$
- S pošle přes AES kanál klientovi A : certifikát identity Ica, který je ve tvaru $(A, V_a)S_s$
- Skype podporuje 2 délky klíče s modulem 1536, 2048 (pro „enterprise“ aplikace)

Přihlášení z pohledu klienta

1. vygeneruje 1024-bitový veřejný a privátní klíč (K_A^+, K_A^-) – jednorázový pár klíčů pro dané sezení
2. vygeneruje 256-bitový symetrický klíč (K) pro AES
3. zašifruje K_A^+, uživatelské jméno a hash hesla pomocí klíče K
4. zašifruje K veřejným klíčem Skype Login Serveru
5. zašifrovaný K_A^+, uživatelské jméno, hash hesla a zašifrovaný klíč sezení K jsou odeslány na Skype Login Server

Přihlášení z pohledu Skype login serveru

1. pomocí privátního klíče dešifruje zprávu a získá K
2. K použije pro dešifrování uživatelského jména, hashe hesla a K_A^+
3. pokud je nalezena shoda pro uživatelské jméno a hash hesla, klient je autentizován
4. vytvoří podepsaný certifikát CA obsahující uživatelské jméno a K_A^+
5. zašle CA klientovi

Vzájemné ověření klientů

- klienti si vymění certifikáty (C_A)
- pro ověření použijí protokol výzva-odpověď (8 bytů):
  1. $A \rightarrow B : R_1$ (8 bytů)
  2. $B \rightarrow A : K_B^-(R_1)$
  3. $A \rightarrow B$: ověří, že $K_B^+(K_B^-(R_1)) == R_1$

Vzájemné ověření klientů

- klienti si vymění certifikáty (C_A)
- pro ověření použijí protokol výzva-odpověď (8 bytů):
  1. $A \rightarrow B : R_1$ (8 bytů)
  2. $B \rightarrow A : K_B^-(R_1)$
  3. $A \rightarrow B$: ověří, že $K_B^+(K_B^-(R_1)) == R_1$

P2P výměna klíčů

- Používá vlastní proprietární systém
- Algoritmus je symetrický
- Proti útoku přehráním je chráněn 64-bit jednorázovým číslem (nonce – number used once)

P2P výměna klíčů 2

- klienti si ustanoví společný 256-bitový klíč sezení K_s (AES) pro šifrování vzájemné komunikace
  1. každý z klientů vygeneruje 128 bitů
  2. vzájemně si své příspěvky podepsané veřejným klíčem protějšku vymění
  3. kryptograficky bezpečnou kombinací obou 128-bitových bloků získá každá ze stran sdílený 256-bitový klíč sezení K_s

Šifrování komunikace

- Všechna komunikace mezi klienty je šifrována XORováním otevřeného textu s bloky, které jsou generovány pomocí blokové šifry AES.
- AES pracuje v ICM (integer counter mode) módu
- Jako klíč je použit klíč K_s ustanovený oběma stranami
- Relace mezi klienty může obsahovat více paralelních toků
- Čítač pro tento ICM mód pak závisí na daném toku, soli a sledu toků

Bezpečnostní politika ustanovená pro Skype

- používá jedinečné uživatelská jména
- uživatele se autentizují pomocí jména a znalosti hesla
- každý uzel poskytuje ostatním uživatelské jméno a to včetně důkazu
- zprávy se posílají v šifrované podobě

PRNG, Signature padding

- PRNG
  - generátor náhodných čísel
  - Implementace se liší na jednotlivých platformách, ve Windows Skype sbírá bity z několika systémových volání, na tyto bity se za použití soli aplikuje hash funkce SHA-1, je použito pouze 64 bitů z výsledku hash funkce
- zarovnání (Signature padding)
  - Vychází ze standardu ISO 9796-2

AES

- AES
  - Skype používá bloky o velikosti 128 a 256 bitů pro klíč
  - AES pracuje v režimu ICM - Integer Counter Mode
  - Při testování implementace byly použity AES-256 standardní testovací vektory a klíče, které byly následně porovnány s výsledky algoritmu AES implementovaným v Perlu
  - Skype se vyznačuje rychlou implementací AES (v porovnání s jinými rychlými C/C++ implementacemi)

RSA

- RSA
  - Využívá variantu standardního „square-and-multiply“ algoritmu a rovněž chytré algoritmy pro umocňování. Zda je číslo prvočíslo, pak ověřuje Miller-Rabinův test s 25 iteracemi $\rightarrow$ což nám zajišťuje, že číslo není přirozené s $pst < 10^{-16}$
  - Korektně implementuje variantu Montgomeryho metody pro modulární inverzi, která je použita pro generování exponentu při dešifrování (privátní klíč)
  - Kód je navržen tak, aby využíval speciální funkce procesoru pro zvýšení efektivity

SHA-1, RC4

- SHA-1 (revize od Toma Bersona – Anagram Laboratories)
  - Úhledný a čistý kód, kompiluje se bez varovaní a chyb
  - Otestovány obě rozhraní funkce, bez nalezení problému
  - Stejné výsledky Jim Gillogyova testu a testovacích vektorů Toma Bersona jako implementace SHA-1 v Perlu
- RC4
  - Tato proudová šifra je ve Skypu použita pro generování prvočísel pro RSA a je inicializována náhodnými bity

Analýza síťového provozu

- šifrovaná komunikace se jeví jako náhodná data
- chaos v P2P architektuře
  - mnoho peerů a nejasná identifikace cíle
- vytváření síťového provozu i v době nečinnosti
  - směrování, pingy

$\rightarrow$ není možné rozeznat legitimní chování od potenciálně škodlivého (šifrovaná spojení na neobvyklých portech, aktivita během nečinnosti...)

Analýza programu

- mnoho nejrůznějších ochran
- využití triků znesnadňujících (znemožňujících) debugging
- využití obfuskátoru k zatemnění kódu nebo šifrování

→ **vyvolává „konspirační otázky“**

- proč firma nepůsobící v Open Source poskytuje zadarmo program, který funguje bezvadně a vyvíjí tolik úsilí, aby nešel reverzovat?
- je snad co skrývat?

→ **nemožnost otestovat na přítomnost škodlivého software**

Ochranné mechanizmy

- znemožnění statického disassemblování
  - šifrování binárky
    - na některé části binárky je aplikován XOR s pevně daným klíčem
    - dešifrování probíhá za běhu v paměti
  - změna struktury binárky za běhu
    - smazání začátku kódu
    - dešifrování zašifrovaných oblastí kódu
    - nahrazení části WinAPI import tabulky (PE exe)

Kontrola integrity kódu

- systém kontroly integrity sestává z několika jednotek počítajících kontrolní součty
  - každá jednotka trochu jiná = jsou polymorfní?
  - spouštěny náhodně
  - podmínky cyklů mají náhodná znaménka
  - použití náhodných operátorů (add, xor, sub. . .)
  - náhodná délka kontrolního součtu
    $\rightarrow$ finální kontrola integrity velmi netriviální úlohou
  - finální kontrolní součet se využívá k výpočtu ukazatele na další úsek kódu

Ochrana citlivého kódu

- použití obfuskátoru
  - účelem je znemožnění (znesnadnění) zpětného inženýrství
    - vytváří nepříjemný chaos v kódu
- klady
  - rapidně zpomaluje studii kódu
  - znemožňuje přímé vykradení kódu
- zápory
  - citelné zpomalení chodu aplikace
  - nárůst velikosti aplikace

Maskování síťového provozu

- Skype nad UDP
  - rámce mohou být několika typů:
    - maskující (obfuskační) data
    - potvrzující (ACK/NACK) paket
    - přeposílána data
    - posílaná data
  - využití extra obfuskační vrstvy
    - zmatek v hlavičce paketu
    - šifrování obsahu paketu

Síťová obfuskační vrstva

- šifruje data proudovou RC4 šifrou
- klíč je generován z prvků datagramu
  - zdrojová a cílová IP adresa
  - ID Skype paketu
  - IV (inicializační vektor) obfuskační vrstvy
- tato RC4 šifra je stavebním kamenem obfuskace na síti
  - používá 80B (80 bajtů) klíč
  - není jasné, kde se bere seed
    - patrně věštba z hvězd
- šifra se používá POUZE k obfuskaci, nikoli za účelem dosažení soukromí!

Maskování síťového provozu 2

- Skype nad TCP
  - RC4 seed je poslán v prvních 4 bajtech proudu
  - RC4 proud pak dešifruje následujících 10 bajtů
  - následně je RC4 proud reinicializován a použit k dešifrování zbytku proudu
- toto dvojité použití stejného proudu umožňuje detekci Skype provozu bez nutnosti vypořádávat se s deobfuskací
- otázkou je, k čemu je to dobré vedět, když stejně není možné Skype blokovat?

Přenos dat na nižších vrstvách

- temeř vše zašifrováno
- data mohou být fragmentována
- komprese každého paketu
  - využití aritmetického komprimování
  - algoritmus blízký Huffmanovu kódování
    - nač používat zbytečně jednoduchý zip

Pod pokličkou Skype – shrnutí

- plusy
  - dobře promyšlený systém
  - hezké použití kryptografie
- mínusy
  - temeř nemožné prosadit bezpečnostní politiku při použití Skype
  - ze síťového provozu nemůžeme určit, jedná-li se o legitimní přenos nebo únik dat
  - nekompatibilní s IDS a monitory síťového provozu
  - úplný black-box, zcela chybí alespoň jakási transparentnost protokolu

Útoky na protokol

- Man-in-the-middle.
- Replay atack.
- Hádání hesla.
- Slabiny v používání CRC.
- Útok postranním kanálem.
- Útok na ASN1.

Man-in-the-Middle

- útočník musí přesvědčit obě strany, že komunikují přímo mezi sebou
- útočník musí zabránit přímé komunikaci mezi těmito dvěma stranami

Replay atack

- útočník může odposlechnout několik handshaků s cílovým uzlem
- může pak zaslat výzvu předstírajíc, že je některý z předchozích účastníků
- pokud uzel zašle stejnou výzvu, jako v předchozím běhu protokolu, útočník ví, jak odpovědět
- pravděpodobnost při N odposlechnutích je však pouze $N / 2^{64}$

Hádání hesla

- uživatele mají možnost zapamatování hesla – heslo je „bezpečně“ uloženo operačním systémem
- útočník se může snažit uhodnout heslo např. slovníkovými útoky
- autentizační centrála Skype vynucuje timeout po několika neúspěšných pokusech o přihlášení

Slabiny při používání CRC

- CRC kontrolní součty:
  - lineární hashovací funkce
  - vhodné pro odhalení náhodných bitových chyb
  - nevhodné pro obranu před úmyslnými změnami
- Podobné používání CRC jako ve WEP
  - oprava naplánována

Slabiny při používání CRC

- CRC kontrolní součty:
  - lineární hashovací funkce
  - vhodné pro odhalení náhodných bitových chyb
  - nevhodné pro obranu před úmyslnými změnami
- Podobné používání CRC jako ve WEP
  - oprava naplánována

Útok postranním kanálem

- založeno na monitorování využívání sdílených prostředků (HDD, CPU):
- z nichž mezi nejběžnější patří:
  - **Timing attack** – měření času CPU
    (výpočet různých operací trvá různě dlouho)
  - **Power monitoring attack** – měření spotřeby energie CPU
    (různé operace mají různou energetickou náročnost)
- Skype se proti těmto útokům nijak nebrání

Útok na ASN1

- Skype nepoužívá přímo ASN1
- avšak používá podobné mechanismy:
  - spoléhá na korektní parsování zakódovaných dat (payloadu)
  - útočník může libovolné upravit některá pole
  - objeveny potenciální chyby v dekódování integerů
  - nehrozí však narušení důvěrnosti komunikace

Závěr

- Multiplatformní, široce používáná close-source aplikace.
- Využívá silnou kryptografii, jako AES, RSA, RC4, SHA-1.
- Bez dokumentace aplikace či protokolu, schopná projít většinou bezpečnostních omezení.
- Robustní, decentralizovaná struktura sítě, využívající zdroje klientů

- Důvěryhodnost a bezpečnost aplikace je neprokazatelná

Zdroje

- Skype Security Evaluation, T Berson, October 2005
- Silver Needle in the Skype, P Biondi, F Desclaux - BlackHat Europe, 2006
- An analysis of the skype peer-to-peer internet telephony protocol, SA Baset, H Schulzrinne - IEEE infocom, 2006
- Skype Detection: Traffic Classification In the Dark, A Nucci, Narus, 2006
- Skype, Wikipedia EN
- Skype Security, Wikipedia EN