Exception Triggered DoS Attacks on Wireless Networks

Yao Zhao, Sagar Vemuri, Jiazhen Chen, Yan Chen, Hai Zhou
Lab for Internet and Security Technology (LIST), Northwestern Univ., USA

Judy (Zhi) Fu
Motorola Labs, USA
Motivation and Contributions

• Proactively search for vulnerabilities in emerging wireless network protocols

• Model checking of protocols?
  - Found an initial ranging vulnerability in WiMAX [NPSec 06]
  - However, many challenges encountered, e.g., protocol ambiguity, hard to test all possible inputs (state explosion)

• Our contributions
  - Reveal a family of exception triggered DoS attacks across many protocols (fast and easy!)
  - Demonstrate feasibility by real experiments
  - Propose countermeasures
Basic Idea

• Processing error messages imprudently
  - Error messages before authentication in clear text
  - Messages are trusted without integrity check

• Vulnerabilities received little attention
  - Not practical in wired network (e.g. TCP reset)
  - Wireless links encrypted at layer 2
Attack Framework

• Attack Requirements
  - Media: sniff and spoof packets
  - Protocol: existence of fatal error conditions before encryption starts
  - Timing: existence of time window to allow injection of faked packets b4 normal packets

• Attack Methodology
  Spoof and inject:
  - error messages that directly trigger exception handler
  - misleading messages that indirectly trigger exception handler
Attack Properties

• Easy to Launch: No need to change MAC
  - Only commodity hardware needed
• Efficient and Scalable:
  - Small attack traffic, attack large # of clients
• Stealthy
  - Can’t be detected w/ current IDS
• Widely Applicable to Many Protocols
Outline

• Motivation
• Attack Framework
• Attack Case Studies
  - TLS based EAP protocols
  - Mobile IPv6 routing optimization protocol
• Countermeasures
• Conclusions
EAP Authentication on Wireless

EAP Layer

- EAP-TLS
- EAP-TTLS
- PEAP
- EAP-FAST
- EAP-SIM
- EAP-AKA

Extensible Authentication Protocol (EAP)

EAP Over LAN (EAPOL)

Data Link Layer

- 802.11 WLAN
- GSM
- UMTS/CDMA2000

Authentication primitive

Authentication method layer

Challenge/Response
TLS Authentication Procedure

TLS Handshake Protocol

Client and Server negotiate a stateful connection

- Mutual authentication
- Integrity-protected cipher suite negotiation
- Key exchange

Client End

Hello Request

Client Hello

Server Hello

Server Certificate

Key-exchange message

Server Hello Done

Client Key-exchange message

Change cipher Spec

TLS finished

Change cipher Spec

TLS finished

Encrypted conversation over TLS

Server End
TLS-based Vulnerability

• Sniff to get the client MAC addr and IDs
  - Packet in clear text before authentication

• Immediately send spoofed error/misleading messages
  - E.g., attacker spoofs an alert message of level 'fatal', followed by a close notify alert.
  - Then the handshake protocol fails and needs to be tried again.

• Complete DoS attack
  - Repeats the previous steps to stop all the retries

• When this attack happens, WPA2 and WPA are all in clear text.
Error Message Attack on TLS: Attacker Spoofing as Server

Client End → Attacker → Server End

Attack Point-1

- Hello Request
- Client Hello
- Error Message
- Server Hello
- Server Certificate
- Server Key-exchange message
- Certification Request
- Server Hello

Attack Point-2

- Certificate
- Client Key-exchange message
- Certificate Verify
- Finished
- Error Message
- Failure
Error Message Attack on TLS: Attacker Spoofing as Client

Attack Point-1

Client

Hello Request

Client Hello

Server Hello

Server Certificate

Server Key-exchange message

Certificate Request

Server Hello Done

Failure

Certificate

Client Key-exchange message

Certificate Verify

Finished

Error msg

Failure

Error msg

Attack Point-2

Client

Attacker

Server
Misleading Message Attack on TLS

Client → Attacker → Server

Trigger → Spooed Server Hello w/ misleading info → Client Hello → Server Hello → Error msg → Failure

Server Certificate → Server Key-exchange message → Certificate Request → Server Hello Done → Certificate → Certificate Verify → Finished
DoS Attack on Challenge/Response over EAP-AKA

- Authentication in UMTS/CDMA2000
- Pre-shared key (Ki) in SIM and AuC
- Send Error Rejection or Notification message

Client End

EAP-Request/Identity

EAP-Response/Identity (NAI)

AKA-Challenge (RAND, AUTN, MAC)

AKA-Response (RES, MAC)

AKA-Authentication-Reject

AKA-Notification

Server End

EAP-Success
Experiments on PEAP WiFi Networks

• Feasibility test on net management utilities
  - Windows native client (XP and Vista)
  - Dell utility - Proxim Utility,
  - the Linux Network Manager of Ubuntu

• Attacker Hardware
  - Wifi cards with Atheros chipsets (e.g. Proxim Orinoco Gold wireless adapter)

• Attacker Software
  - Libraries : Libpcap (sniffing) & Lorcon (spoofing)
  - MADWifi driver to configure CWMin
  - Attacking code: 1200 lines in C++ on Ubuntu Linux
Field Test Results

Conducted EAP-TLS attacks at a major university cafeteria

- 2 Channels, 7 Client Hosts in all, and 1 Attacker
- Successfully attacked all of them in one channel
### Attack Efficiency Evaluation

<table>
<thead>
<tr>
<th>Attack Point 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio by # of Messages</td>
<td>25.00% [1/4]</td>
</tr>
<tr>
<td>Ratio by Bytes</td>
<td>15.89% [78/491]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attack Point 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio by # of Messages</td>
<td>28.57% [2/7]</td>
</tr>
<tr>
<td>Ratio by Bytes</td>
<td>14.87% [156/1049]</td>
</tr>
</tbody>
</table>

- For example, when attack happens at the second point
  - Just need to send 156 bytes of message to screw the whole 1049 bytes authentication messages.
Attack Scalability Evaluation

• NS2 Simulation Methodology
  - One TLS-Server and one base station
    • 100MBps duplex-link between BS and TLS-Server with various delay
  - 1~50 TLS-Clients
    • Poisson inter-arrival (avg 0.5s)
    • Retry at most 18 times with the interval of 1s
  - One TLS-Attacker
  - All results are based on an average of 20 runs

• Simulation Results
  - Attackers can reduce CWMin to be aggressive
  - Attacks very scalable: all clients fail authentications
Time Window Sensitivity w/ Various Server Delay

- Attack succeed even with very small time window
- The larger the server delay, the larger chance for attack messages to reach victim client before legitimate message.
Outline

• Motivation
• Attack Framework
• Attack Case Studies
  - TLS based EAP protocols
  - Mobile IPv6 routing optimization protocol
• Countermeasures
• Conclusions
Mobile IPv6 Protocol

- Allows a mobile node (MN) to remain reachable while moving in the IPv6 Internet.
  - A MN is always identified by its home address, regardless of its current point of attachment.
  - IPv6 packets addressed to a MN's home address are transparently routed to its care-of address.
  - The protocol enables IPv6 nodes to cache the binding and thus to send any packets destined for the MN directly to it.
Return Routability Procedure
Bind Error Vulnerability

The Binding Error message is not protected.
Bind Acknowledgement Vulnerability

Binding Acknowledgement is not protected either
Attack Power and Evaluations

• The attack can also disrupt on-going sessions
  - RR procedure repeats every few minutes

• Emulation experiments
  - Build the mobile IPv6 network using the Mobile IPv6 Implementation for Linux (MIPL v2.0).
  - GRE-based (Generic Routing Encapsulation) interfaces tunnel IPv6 over IPv4
  - Conducted 100 times.
  - All RR request failed - performance degradation attack
Outline

• Motivation
• Attack Framework
• Attack Case Studies
  - TLS based EAP protocols
  - Mobile IPv6 routing optimization protocol
• Countermeasures
• Conclusions
Countermeasures

• Detection: Based on Two Symptoms
  - Conflict messages and abnormal protocol end

• Protocol Improvement (band-aid fix)
  - Wait for a short time for a success message (if any) to arrive
  - Accept success messages over errors/failures
  - Start multiple session for multiple responses (for misleading message attack)
  - Implemented and repeated attack experiments: all attacks failed.

• Design of Robust Security Protocols
  - Get packets encrypted and authenticated as early as possible.
Conclusions

• Propose exception triggered denial-of-service attacks on wireless sec protocols
  – Explore the vulnerabilities in the exception handling process

• Demonstrate attack effects
  – TLS based EAP protocols
    • Real-world experiments and simulations
  – The Return Routability procedure of Mobile IPv6 protocol
    • Testbed emulations

• Propose detection scheme and protocol improvement principle
  • Real implementation and experiments

• Working with IETF on improving protocol standards
Backup Slides
Case Study 1:
Attack on TLS based EAP Protocols in Wireless Networks
EAP and TLS Authentication

• **Transport Layer Security (TLS)**
  - Mutual authentication
  - Integrity-protected cipher suite negotiation
  - Key exchange

• **Challenge/Response authentication in GSM/UMTS/CDMA2000**
  - Pre-shared key (Ki) in SIM and AuC
  - Auc challenges mobile station with RAND
  - Both sides derive keys based on Ki and RAND
Other Related Work

- Many DoS Attacks on Wireless Net
  - Jamming, Rogue AP, ARP spoofing
  - More recent: deauthentication and virtual carrier sense attacks [Usenix Sec 03]
For the 33 different tries
- All suffered an attack at Attack Point-1
- 21% survive from the first attack but failed at the 2nd Attack Point.
• The lower $CWMin$ of the attacker, the higher attack success ratio.

• Attack is scalable: very few clients are able to authenticate successfully.
Vulnerabilities of RR Procedure

• Binding Error Vulnerability
  - Mobile node SHOULD cease the attempt to use route optimization if the status field is set to 2 (unrecognized Mobility header) in Binding Error message.
  - The Binding Error message is not protected.

• Bind Acknowledgement Vulnerability
  - Binding Acknowledgement with status 136, 137 and 138 is used to indicate an error
  - Binding Acknowledgement is not protected either
PEAP Enhancement

• Original WPA supplicant v0.5.10
  - Generate TLS ALERT on unexpected messages
  - Stop authentication on TLS ALERT

• Delayed response implementation
  - Drop unexpected message silently
  - Wait for 1 second when receiving TLS ALERT to allow multiple responses, and ignore TLS ALERT response if good responses received
  - Multiple sessions against misleading messages

• Verification
  - Repeated the WiFi attack experiments
  - All attacks failed
Design of Robust Security Protocol

Client

Server certificate (including server’s public key $K_s^+$)

$K_s^+$ (Random string $S$, Identity)

Server

No useful info

Server ignore error msg. New session with secret $S$ and $ID$

Cannot be spoofed