DNSSEC

DNSSEC Protocol

- Client uses DNS to get **www.google.com's IP address**
- **Goal**
  - Extend DNS so that client gets a ***verifiable*** answer for www.google.com's IP address
  - **Integrity & Authenticity**, not Confidentiality/Availability
  - **Object security**, not channel security
- **Design**
  1. Have google.com return and ***sign*** the final answer
  2. Verify that signature via a chain of keys & signatures that is anchored in **KSK_{root}**
  3. Client and resolver start off w/ a trust anchor = DNS root's **KSK_{root}** (Root = ".")

DNSSEC Resource Records (RRs)

- **RR** (Resource record) = basic unit of data in DNS
- **An *"A Record"***: contains IP address for a domain
- **Zone**: administrative region of domain name space (e.g., ".com" zone or "google.com" zone)

- New RRs in DNSSEC:
  - ***RRSIG***: signature by a zone over one of its RR sets (e.g. signature over the zone's A records)
  - ***DNSKEY***: a public key to verify a zone's RRSIGs (either KSK or ZSK)
    - Zone uses ZSK for RRSIG's of all its data RRs, but *not* its DNSKEY RRs
    - Zone uses **KSK** for RRSIG's of *its own* DNSKEY records
  - ***DS (Delegation of Signing)***: statement (hash) of a KSK-DNSKEY record for one of zone's *children*

DNSSEC Protocol

- Client and resolver start off knowing root's **KSK_{root}** (trust anchor)

1. Client asks resolver for www.google.com's IP address
2. Resolver issues recursive queries:
3. From root zone:
    1. DNSKEY: Root's ZSK = ZSK_{root}
    2. RRSIG on DNSKEY using **KSK_{root}**
    3. DS: statement (hash) of $KSK_{.com}$
    4. RRSIG on DS using ZSK_{root}

DNSSEC Protocol (cont.)

- Next from .com zone:
  1. DNSKEY: .com's KSK = $KSK_{.com}$
     1. Previously got DS = hash of $KSK_{.com}$ from root (.com's parent)
  2. DNSKEY: .com's ZSK = $ZSK_{.com}$
  3. RRSIG on DNSKEY using $KSK_{.com}$
  4. DS: statement (hash) of $KSK_{google.com}$
  5. RRSIG on DS using $ZSK_{.com}$

DNSSEC Protocol (cont.)

- Finally from google.com zone:
  1. DNSKEY: google.com's KSK = $KSK_{google.com}$
     1. Previously got DS = hash of $KSK_{google.com}$ from .com (google.com's parent)
  2. DNSKEY: google.com's ZSK = $ZSK_{google.com}$
  3. RRSIG on DNSKEY using $KSK_{google.com}$
  4. RR (A Record): www.google.com -> 1.1.1.1
     - Final answer!
  5. RRSIG on RR using $ZSK_{google.com}$

NSEC(3): DNE in DNSSEC

- **NSEC(3)**: Secure way to let client know that domain query **does not exist (DNE)**
- For query that DNE:
  - Name server ***hashes*** (NSEC3) the query
  - Gets the "sandwiching" hashes that do exist
  - Returns this "sandwich" of hashes as the final answer with an RRSIG

- Slows down ***zone enumeration*** attacks by forcing lots of guessing

Spring 2016 Problem 12

Sarah is using a laptop on her home network to browse the Internet over HTTP. Consider an adversary with a smartphone and an Internet connection and the following capabilities:

A. No additional capabilities
B. A $10k GPU compute cluster
C. Can compromise Sarah's home router
D. Can compromise the primary nameserver for the zone
E. Can compromise the primary nameserver and offline signing key for the zone

(a) In the following table, **mark with an X** which attacks each adversary could successfully perform against Sarah if everyone uses **DNS without DNSSEC**:

	A	B	C	D
Spoof existence of records that actually don't exist in the zone				
Spoof values of records that exist in the zone				
Spoof non-existence of records that actually do exist in the zone				
Enumerate all names in the zone				

Spring 2016 Problem 12

Sarah is using a laptop on her home network to browse the Internet over HTTP. Consider an adversary with a smartphone and an Internet connection and the following capabilities:

A. No additional capabilities
B. A $10k GPU compute cluster
C. Can compromise Sarah's home router
D. Can compromise the primary nameserver for the zone
E. Can compromise the primary nameserver and offline signing key for the zone

(a) In the following table, **mark with an X** which attacks each adversary could successfully perform against Sarah if everyone uses **DNS without DNSSEC**:

	A	B	C	D
Spoof existence of records that actually don't exist in the zone			X	X
Spoof values of records that exist in the zone			X	X
Spoof non-existence of records that actually do exist in the zone			X	X
Enumerate all names in the zone				X

Spring 2016 Problem 12

Sarah is using a laptop on her home network to browse the Internet over HTTP. Consider an adversary with a smartphone and an Internet connection and the following capabilities:

A. No additional capabilities
B. A $10k GPU compute cluster
C. Can compromise Sarah's home router
D. Can compromise the primary nameserver for the zone
E. Can compromise the primary nameserver and offline signing key for the zone

(b) Remember that when using DNSSEC, the zone is signed with a key that is not stored on the primary nameserver. Instead, the key is stored at an "offline" location. Also, NSEC records sign each adjacent pair of names in the zone. In the following table, **mark with an X** which attacks each adversary could successfully perform against **DNSSEC with NSEC records**:

	A	B	C	D	E
Spoof existence of records that actually don't exist in the zone					
Spoof values of records that exist in the zone					
Spoof non-existence of records that actually do exist in the zone					
Enumerate all names in the zone					

Spring 2016 Problem 12

Sarah is using a laptop on her home network to browse the Internet over HTTP. Consider an adversary with a smartphone and an Internet connection and the following capabilities:

A. No additional capabilities
B. A $10k GPU compute cluster
C. Can compromise Sarah's home router
D. Can compromise the primary nameserver for the zone
E. Can compromise the primary nameserver and offline signing key for the zone

(b) Remember that when using DNSSEC, the zone is signed with a key that is not stored on the primary nameserver. Instead, the key is stored at an "offline" location. Also, NSEC records sign each adjacent pair of names in the zone. In the following table, **mark with an X** which attacks each adversary could successfully perform against **DNSSEC with NSEC records**:

	A	B	C	D	E
Spoof existence of records that actually don't exist in the zone					X
Spoof values of records that exist in the zone					X
Spoof non-existence of records that actually do exist in the zone					X
Enumerate all names in the zone	X	X	X	X	X

Spring 2016 Problem 12

Sarah is using a laptop on her home network to browse the Internet over HTTP. Consider an adversary with a smartphone and an Internet connection and the following capabilities:

A. No additional capabilities
B. A $10k GPU compute cluster
C. Can compromise Sarah's home router
D. Can compromise the primary nameserver for the zone
E. Can compromise the primary nameserver and offline signing key for the zone

(c) NSEC3 records were introduced to fix certain issues with NSEC records: each name in the zone is hashed, the hashes are sorted, and each NSEC3 record signs an adjacent pair of hash digests. In the following table, **mark with an X** which attacks each adversary could successfully perform against **DNSSEC with NSEC3 records**:

	A	B	C	D	E
Spoof existence of records that actually don't exist in the zone					
Spoof values of records that exist in the zone					
Spoof non-existence of records that actually do exist in the zone					
Enumerate all names in the zone					

Spring 2016 Problem 12

Sarah is using a laptop on her home network to browse the Internet over HTTP. Consider an adversary with a smartphone and an Internet connection and the following capabilities:

A. No additional capabilities
B. A $10k GPU compute cluster
C. Can compromise Sarah's home router
D. Can compromise the primary nameserver for the zone
E. Can compromise the primary nameserver and offline signing key for the zone

(c) NSEC3 records were introduced to fix certain issues with NSEC records: each name in the zone is hashed, the hashes are sorted, and each NSEC3 record signs an adjacent pair of hash digests. In the following table, **mark with an X** which attacks each adversary could successfully perform against **DNSSEC with NSEC3 records**:

	A	B	C	D	E
Spoof existence of records that actually don't exist in the zone					X
Spoof values of records that exist in the zone					X
Spoof non-existence of records that actually do exist in the zone					X
Enumerate all names in the zone		X		X	X

Fall 2016 Problem 6

While browsing memes late at night, you make a DNSSEC-enabled request for the `A` record of `oskibear2k16.berkeley.edu`.

To your dismay and disappointment, you receive the response: `NXDOMAIN`. The response also contains `NSEC` and `RRSIG` records.

(a) What further DNS queries, if any, must you make to validate this response? Assume all caches are empty. Provide both the domain(s) to be queried and the necessary record type(s) for each domain. You don't need to use all the lines below:

Fall 2016 Problem 6

While browsing memes late at night, you make a DNSSEC-enabled request for the `A` record of `oskibear2k16.berkeley.edu`.

To your dismay and disappointment, you receive the response: `NXDOMAIN`. The response also contains `NSEC` and `RRSIG` records.

(a) What further DNS queries, if any, must you make to validate this response? Assume all caches are empty. Provide both the domain(s) to be queried and the necessary record type(s) for each domain. You don't need to use all the lines below:

Solution:

1. the `DNSKEY` record for `berkeley.edu`
2. the `DS` record for `berkeley.edu`
3. the `DNSKEY` record for `.edu`
4. the `DS` record for `.edu`
5. the `DNSKEY` record for `.` (the root)

Fall 2016 Problem 6

(b) What parties must you trust in order to validate this response and confirm that `oskibear2k16.berkeley.edu` really doesn't exist? Circle all that apply and provide no explanation.

1. your local network administrator
2. the DNS server that gave you this response
3. one or more certificate authorities
4. the DNS root zone administrator

Fall 2016 Problem 6

(b) What parties must you trust in order to validate this response and confirm that `oskibear2k16.berkeley.edu` really doesn't exist? Circle all that apply and provide no explanation.

1. your local network administrator
2. the DNS server that gave you this response
3. one or more certificate authorities
4. the DNS root zone administrator

Solution: the DNS root zone administrator

Fall 2016 Problem 6

(c) The devious students at a certain university to the south decide that this will not do. They have found a way to spoof responses to any of your DNS requests, and furthermore, have stolen the private ZSK of the `.edu` zone!

Now when you send a request for `oskibear2k16.berkeley.edu`, you get a response with an A record of `171.64.64.64` and an `RRSIG` that your resolver says is valid.

How did this happen? Be specific. (Once again, assume that all caches are empty.)

Fall 2016 Problem 6

(c) The devious students at a certain university to the south decide that this will not do. They have found a way to spoof responses to any of your DNS requests, and furthermore, have stolen the private ZSK of the `.edu` zone!

Now when you send a request for `oskibear2k16.berkeley.edu`, you get a response with an A record of `171.64.64.64` and an `RRSIG` that your resolver says is valid.

How did this happen? Be specific. (Once again, assume that all caches are empty.)

Solution:

The RRSIG is signed with a fake key, which is given in the spoofed response for the berkeley.edu DNSKEY. A spoofed DS record is sent using the stolen .edu ZSK.

Note that this requires the collusion of neither the berkeley.edu zone admin nor the root zone admin.

Detection Styles, Evasion, NIDS vs. HIDS

Styles of Detection

Needn't be exclusive

- Signature-based: look for activity that matches a known attack (or known malware) Blacklisting
  - \+ Simple; easy to share; addresses a very common threat
  - – Misses novel attacks or variants; can have high FP
- Vulnerability signatures: look for activity that matches a known vulnerability (i.e., how not what)
  - \+ ~Simple; easy to share; addresses v. common threat; detects variants
  - – Misses novel attacks; significant work to develop
- Specification-based: define what activity is okay, flag anything else Whitelisting
  - \+ Can detect novel attacks; possibly low FP
  - – Lots of work; not shareable; churn requires maintenance

Styles of Detection, con't

- Anomaly-based: build up / infer profile of "normal" activity, flag deviations as potential attacks
  - \+ Can detect novel attacks
  - – Can miss both known and novel attacks; training data might be tainted; Base Rate Fallacy can lead to high FPs
- Behavioral: look for specific evidence of compromise rather than attacks themselves
  - \+ Can detect novel attacks; often low FPs; can be cheap
  - – Post-facto detection; narrow, and thus often evadable
- Honeypots: provide system/resource that isn't actually used otherwise, monitor access to it
  - \+ Can detect novel attacks; examine attacker goals
  - – Attacker may spot fakery; noise from *endemic attacks*

The Problem of Evasion

- Most detection approaches can be eluded
  - *Doesn't mean the approach is worthless*
- Evasions arise from uncertainties/ambiguities
  - One strategy to address: impose an interpretation ("normalization")
- Evasion considerations:
  - Incomplete analysis: detector doesn't fully analyze
  - Spec deviations: not all systems implemented correctly
  - Attacker can stress the monitor
    - Exhaust its resources (state, CPU)
    - Exploit its own bugs (crash, code injection)
  - Monitor lacks sufficient information to disambiguate
    - And can't alert on presence of ambiguity due to FPs

NIDS vs. HIDS

- NIDS benefits:
  - Can cover a lot of systems with single deployment
    - Much simpler management
  - Easy to "bolt on" / no need to touch end systems
  - Doesn't consume production resources on end systems
  - Harder for an attacker to subvert / less to trust
- HIDS benefits:
  - Can have direct access to semantics of activity
    - Better positioned to block (prevent) attacks
    - Harder to evade
  - Can protect against non-network threats
  - Visibility into encrypted activity
  - Performance scales much more readily (no chokepoint)
    - No issues with "dropped" packets

Spring 2010 – Problem 4

You work for a company that sells intrusion detection systems. When you start at the company, the core technology is a signature-based scheme we'll call *S*. A little later the company acquires a competitor whose core technology is an anomaly-based scheme we'll call *A*.

Suppose that *S* is a network-based scheme that works by passively analyzing individual UDP and TCP packets. Suppose that *A* is a host-based scheme that is a component of the browser that processes and analyzes individual URLs before they're loaded by the browser. Scheme *S* operates in a stateless fashion and scheme *A* maintains state regarding URLs it has previously analyzed.

Spring 2010 – Problem 4

You work for a company that sells intrusion detection systems. When you start at the company, the core technology is a signature-based scheme we'll call *S*. A little later the company acquires a competitor whose core technology is an anomaly-based scheme we'll call *A*.

Suppose that *S* is a network-based scheme that works by passively analyzing individual UDP and TCP packets. Suppose that *A* is a host-based scheme that is a component of the browser that processes and analyzes individual URLs before they're loaded by the browser. Scheme *S* operates in a stateless fashion and scheme *A* maintains state regarding URLs it has previously analyzed.

(a) For each of the following, circle your answer or answers.

1. With regard to the general properties of different types of detectors, **circle all** of the following that are correct:

   i. To use a signature-based approach like scheme *S* at a new site first requires access to logs of the site's historical activity in order to train the detector.

   ii. It is possible to design a signature-based detector that has no false negatives.

   iii. Specification-based approaches work well for detecting known attacks but do not work well for detecting novel attacks.

   iv. One appealing property of network-based detectors is that they can provide easier management than host-based detectors.

   v. An attraction of behavioral approaches is that they are especially well-suited to prevent initial system compromises.

   vi. None of the above.

Spring 2010 – Problem 4

You work for a company that sells intrusion detection systems. When you start at the company, the core technology is a signature-based scheme we'll call *S*. A little later the company acquires a competitor whose core technology is an anomaly-based scheme we'll call *A*.

Suppose that *S* is a network-based scheme that works by passively analyzing individual UDP and TCP packets. Suppose that *A* is a host-based scheme that is a component of the browser that processes and analyzes individual URLs before they're loaded by the browser. Scheme *S* operates in a stateless fashion and scheme *A* maintains state regarding URLs it has previously analyzed.

(a) For each of the following, circle your answer or answers.

1. With regard to the general properties of different types of detectors, **circle all** of the following that are correct:

   i. To use a signature-based approach like scheme *S* at a new site first requires access to logs of the site's historical activity in order to train the detector.

   ii. It is possible to design a signature-based detector that has no false negatives.

   iii. Specification-based approaches work well for detecting known attacks but do not work well for detecting novel attacks.

   iv. One appealing property of network-based detectors is that they can provide easier management than host-based detectors.

   v. An attraction of behavioral approaches is that they are especially well-suited to prevent initial system compromises.

   vi. None of the above.

Spring 2010 – Problem 4

You work for a company that sells intrusion detection systems. When you start at the company, the core technology is a signature-based scheme we'll call *S*. A little later the company acquires a competitor whose core technology is an anomaly-based scheme we'll call *A*.

Suppose that *S* is a network-based scheme that works by passively analyzing individual UDP and TCP packets. Suppose that *A* is a host-based scheme that is a component of the browser that processes and analyzes individual URLs before they're loaded by the browser. Scheme *S* operates in a stateless fashion and scheme *A* maintains state regarding URLs it has previously analyzed.

2. Your company wants to deploy a new product that provides intrusion prevention functionality, building upon either *S* or *A* as a foundation. The best candidate scheme for this is (**circle just one**):
   i. Scheme *S*, because signature-based approaches have lower false negative rates than anomaly-based approaches.
   ii. Scheme *S*, because signature-based approaches work in real time and anomaly-based approaches do not.
   iii. Scheme *A*, because anomaly-based approaches have lower false positive rates than signature-based approaches.
   iv. Scheme *A*, because anomaly-based approaches require less state than signature-based approaches.
   v. It is not clear without additional information which of schemes *S* or *A* would work better for intrusion prevention.

Spring 2010 – Problem 4

You work for a company that sells intrusion detection systems. When you start at the company, the core technology is a signature-based scheme we'll call *S*. A little later the company acquires a competitor whose core technology is an anomaly-based scheme we'll call *A*.

Suppose that *S* is a network-based scheme that works by passively analyzing individual UDP and TCP packets. Suppose that *A* is a host-based scheme that is a component of the browser that processes and analyzes individual URLs before they're loaded by the browser. Scheme *S* operates in a stateless fashion and scheme *A* maintains state regarding URLs it has previously analyzed.

2. Your company wants to deploy a new product that provides intrusion prevention functionality, building upon either *S* or *A* as a foundation. The best candidate scheme for this is (**circle just one**):

   i. Scheme *S*, because signature-based approaches have lower false negative rates than anomaly-based approaches.

   ii. Scheme *S*, because signature-based approaches work in real time and anomaly-based approaches do not.

   iii. Scheme *A*, because anomaly-based approaches have lower false positive rates than signature-based approaches.

   iv. Scheme *A*, because anomaly-based approaches require less state than signature-based approaches.

   v. It is not clear without additional information which of schemes *S* or *A* would work better for intrusion prevention.

Spring 2010 – Problem 4

You work for a company that sells intrusion detection systems. When you start at the company, the core technology is a signature-based scheme we'll call *S*. A little later the company acquires a competitor whose core technology is an anomaly-based scheme we'll call *A*.

Suppose that *S* is a network-based scheme that works by passively analyzing individual UDP and TCP packets. Suppose that *A* is a host-based scheme that is a component of the browser that processes and analyzes individual URLs before they're loaded by the browser. Scheme *S* operates in a stateless fashion and scheme *A* maintains state regarding URLs it has previously analyzed.

3. As your company becomes more successful you grow concerned that attackers will try to evade your detectors. **Circle all** of the following that are correct:

   i. Scheme *S* is vulnerable to evasion by attackers who can manipulate the order and timing of the packets they send.

   ii. Scheme *A* is vulnerable to evasion by attackers who can force their traffic to be sent using fragmented packets.

   iii. An attacker can more easily try to exhaust the memory used by scheme *S* than the memory used by scheme *A*.

   iv. For analyzing unencrypted HTTP traffic, scheme *A* is more vulnerable to evasion by attackers that employ URL hex-escape encodings than is scheme *S*.

   v. Given how the different schemes operate, scheme *A* is likely better suited for resisting evasion by imposing a canonical form ("normalization") on the traffic it analyzes than is scheme *S*.

   vi. None of the above.

Spring 2010 – Problem 4

You work for a company that sells intrusion detection systems. When you start at the company, the core technology is a signature-based scheme we'll call *S*. A little later the company acquires a competitor whose core technology is an anomaly-based scheme we'll call *A*.

Suppose that *S* is a network-based scheme that works by passively analyzing individual UDP and TCP packets. Suppose that *A* is a host-based scheme that is a component of the browser that processes and analyzes individual URLs before they're loaded by the browser. Scheme *S* operates in a stateless fashion and scheme *A* maintains state regarding URLs it has previously analyzed.

3. As your company becomes more successful you grow concerned that attackers will try to evade your detectors. **Circle all** of the following that are correct:

   i. Scheme *S* is vulnerable to evasion by attackers who can manipulate the order and timing of the packets they send.

   ii. Scheme *A* is vulnerable to evasion by attackers who can force their traffic to be sent using fragmented packets.

   iii. An attacker can more easily try to exhaust the memory used by scheme *S* than the memory used by scheme *A*.

   iv. For analyzing unencrypted HTTP traffic, scheme *A* is more vulnerable to evasion by attackers that employ URL hex-escape encodings than is scheme *S*.

   v. Given how the different schemes operate, scheme *A* is likely better suited for resisting evasion by imposing a canonical form ("normalization") on the traffic it analyzes than is scheme *S*.

   vi. None of the above.

Malware

Malware Overview

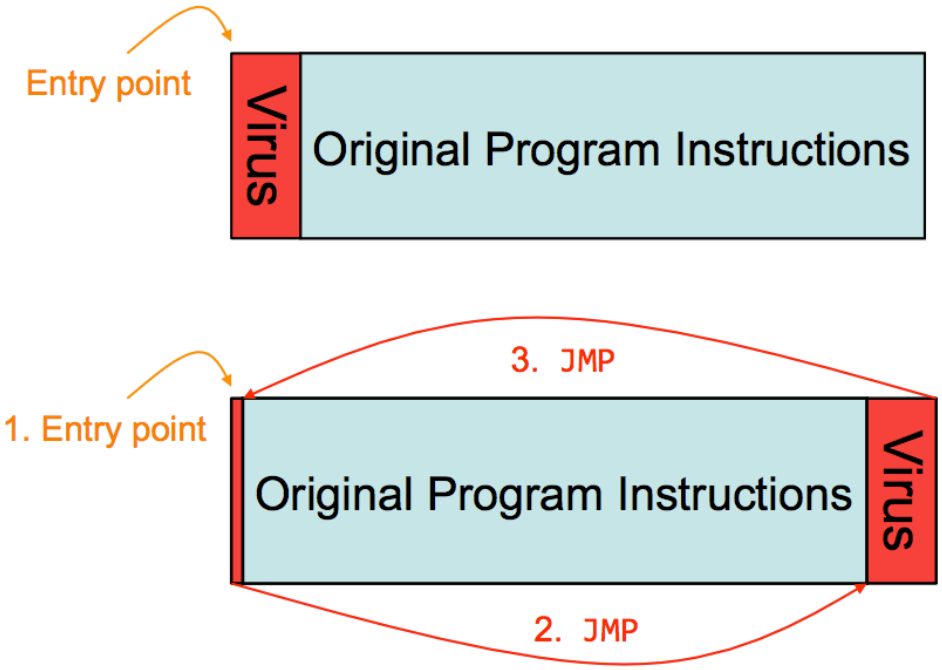
- **Malware**: malicious code that runs on a victim's system
- Rich detection & evasion **arms race**
- **Botnet** architecture: decouple compromising machines and controlling machines
  - Attacker ("**botmaster**") acquires a collection of compromised machines (a "**botnet**")
  - Bot code talks to **C&C** serve to download updates/commands
- Lots of ways to compromise machines:
  - PPI Ecosystem
  - Viruses
  - Worms
  - …

Malware That Automatically Propagates

- Virus = code that propagates (**replicates**) across systems by arranging to have itself *eventually executed,* creating a new additional instance
  - Generally infects by altering stored code
- Worm = code that self-propagates/replicates across systems by arranging to have itself *immediately executed* (creating new addl. instance)
  - Generally infects by altering running code
  - No user intervention required
- (Note: line between these isn't always so crisp; plus some malware incorporates both approaches)

Malware That Automatically Propagates

- Virus = code that propagates (**replicates**) across systems by arranging to have itself *eventually executed,* creating a new additional instance
  - Generally infects by altering stored code



Evading Anti-Virus Detection

- **Polymorphic** code: *Encrypt* main virus code
  - New Virus code =
    - [decryptor code +
    - key +
    - encrypted {main virus code + encryptor code}]
  - **Execute**: decryptor uses key to decrypt & run main virus code
  - **Propagate**:
    - Encryptor generates new key and re-encrypts {main virus code + encryptor code}
    - Infect new host program with
      - [decryptor code +
      - new key +
      - newly encrypted {main virus code + encryptor code}]

Evading Anti-Virus Detection

- **Metamorphic code**: Rewritten but semantically equivalent code
  - **Execution**: runs like normal virus
  - **Propagation**: rewrite code right before infecting new host program
    - Reorder operations
    - Add padding/No-OP's
    - Renumber registers
    - …