Jianxin HUANG, Bo YU, Runhao LIU, Jinshu SU, 2023. Automatic discovery of stateful variables in network protocol software based on replay analysis. *Frontiers of Information Technology & Electronic Engineering*, 24(3):403-416. https://doi.org/10.1631/FITEE.2200275

Automatic discovery of stateful variables in network protocol software based on replay analysis

Key words: Stateful variables; Network protocol software; Program analysis technology; Network security

Corresponding authors: Bo YU, Jinshu SU
E-mail: yubo0615@nudt.edu.cn, sjs@nudt.edu.cn
ORCID: Bo YU, https://orcid.org/0000-0001-6576-5555;
Jinshu SU, https://orcid.org/0000-0001-9273-616X

Background

- The state of a protocol can be measured by a state machine [1].
- Stateful variables: variables or data structures used to hold the information related to the state in network protocol programs have a degree of stateful characteristics.
  - e.g., OpenSSL, a pair of stateful variables with the length constraint
  - A struct *session_id_context* is used to ensure that sessions are reused only in the appropriate context
  - An unsigned integer variable *sid_ctx_length* is used to specify the length of the struct above
- Stateful variables are closely related to:
  - normal operation of protocol services
  - correct processing of protocol data
  - safe execution of the protocol software program

Motivation

- Vulnerable points result from abnormal states of variables due to:
  - ambiguities of specifications
  - software deviations in implementations
  - coding errors or improper uses of stateful variables by program developers

- Previous related work is **insufficient** at present:
  - Most methods for describing and discovering logical errors are concerned with variables in common binary via static analysis [2]
  - Lack effective analysis methods for detecting the transitions of protocol states and conversions of multiple sessions
  - Traditional vulnerability mining approaches may take too much time to generate such inputs to explore a specific state [3]
  - Vulnerabilities possibly appear in many forms, not only a crash, but also privacy disclosure, authentication bypass, RCE, etc. [4]

Motivation

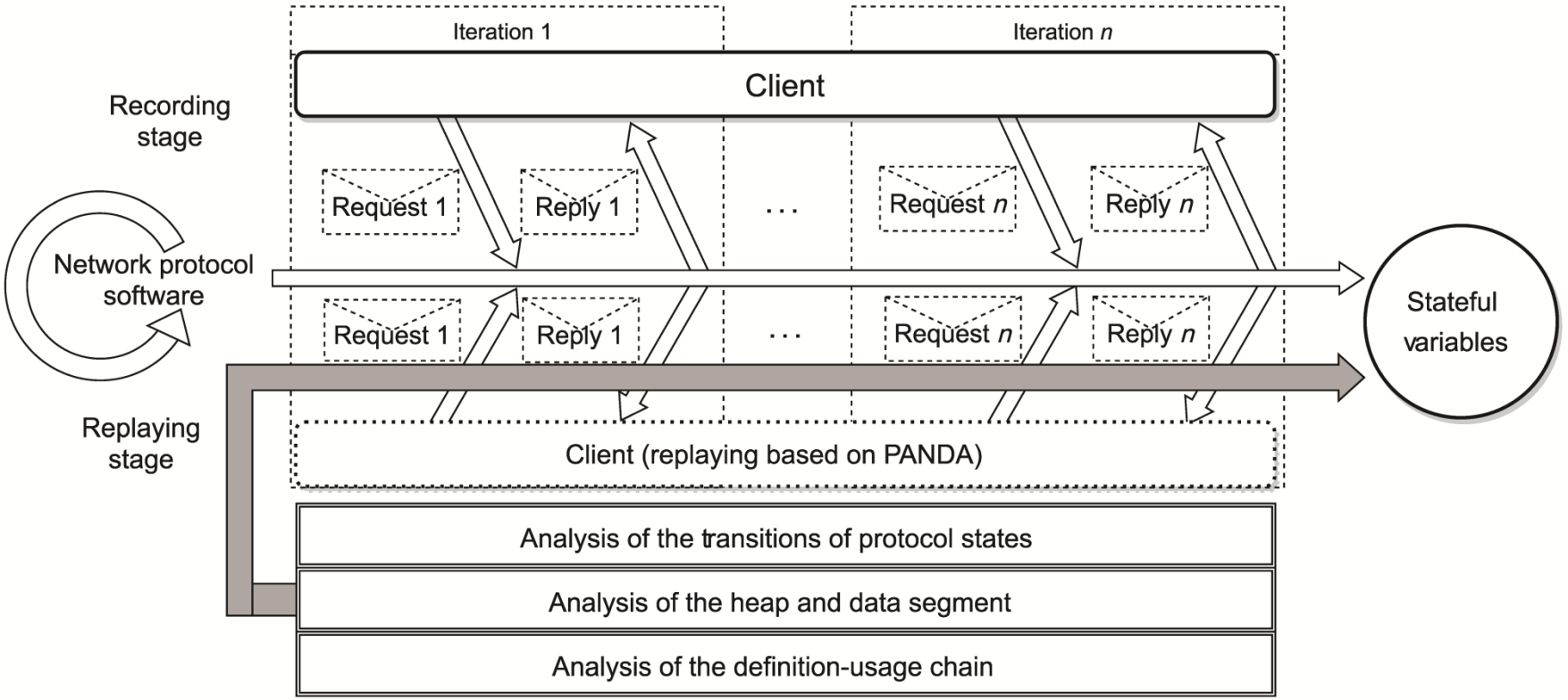
Technological challenges of discovering stateful variables in network protocol software:

- **Low automaticity**. Stateful variables need to be differentiated by analyzing a series of network protocol software behaviors. Extensive manual work is required in testing and verifying a vulnerability caused by misused stateful variables and lack of automatic approaches.
- **Hard to discover**. Stateful variables are closely related to the protocol software states and network inputs. It is necessary to use a specific sequence of data packages to reach a state that can cause errors with stateful characteristics. This conduct makes it difficult to cover deeper states, so the variables mistaken in deeper paths cannot be found eventually.

Main contributions

- An **automatic technique** that is novel and can work on binaries of different protocol software programs is proposed to discover stateful variables.
- A stateful variable **discovery algorithm** is proposed from three dimensions, about timing, spacing, and operating sequence. It is suitable for a deeper analysis of complicated states in protocol software by recording and replaying the executed trace.
- A **prototype system** implemented by our approach to validate our idea is applied to 11 protocol software programs, with an average true positive rate (TPR) of 82% and an average precision up to 96%.

Framework

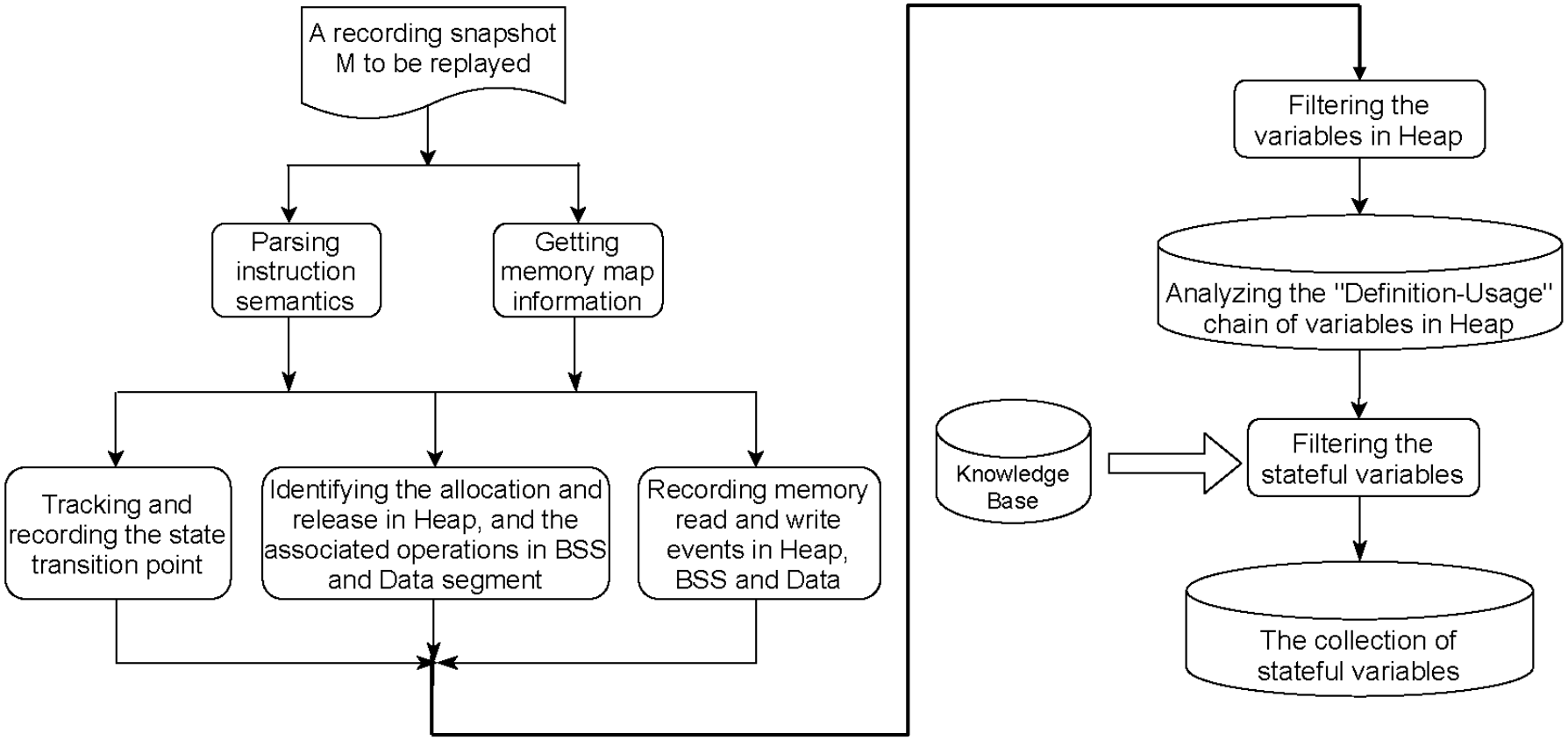


Overview of the prototype system for automatic discovery of stateful variables

- Recording and replaying module
- Address space analysis module

- Protocol state analysis module
- Address space analysis module
- Definition-usage chain analysis module

Flowchart of the algorithm



Flowchart of the automatic discovery algorithm for stateful variables of network protocol software

Evaluation

- The experiments designed for evaluation consist of two parts:
  - Testing in a **benchmark**, ProFuzzBench (9 protocol software programs)
  - Testing in **real-world** programs (2 protocol software programs)
- We obtained the numbers of global variables and variables with stateful characteristics in the source codes by human checking

Protocol software	Number of variables		
	SVDA	GVFM	SVEM
DAAP/forked-daapd	314	446	388
DICOM/Dcmtk	372	593	453
DNS/Dnsmasq	0	288	163
DTLS/TinyDTLS	168	226	201
FTP/LingtFTP	216	379	263
RTSP/Live555	398	580	477
SIP/Kamailio	499	647	593
SMTP/Exim	569	742	681
SSH/OpenSSH	759	962	920
TLS/OpenSSL	738	934	897
SSH/LibSSH	405	591	479

SVDA: stateful variables discovered automatically; GVFM: global variables found manually; SVEM: stateful variables estimated manually

The number of stateful variables automatically discovered by the prototype system compared to the numbers of global and stateful variables estimated by manual inspection from 11 protocol software programs

Evaluation

- To evaluate if our approach can be applied to various protocol software, we introduce some widely used metrics:
  - true positive rate (TPR) or recall, true negative rate (TNR), false positive rate (FPR), false negative rate (FNR), precision (*P*), and F1-measure (F1).

Results of six evaluation metrics for indicating the effectiveness of our prototype system

Protocol software	TPR (%)	TNR (%)	FPR (%)	FNR (%)	*P* (%)	F1 (%)
DAAP/forked-daapd	80.58	89.23	10.77	19.42	97.77	88.35
DICOM/Dcmtk	81.88	95.89	4.11	18.12	98.39	89.38
DNS/Dnsmasq	0.00	100.00	0.00	100.00	NA	NA
DTLS/TinyDTLS	81.87	56.82	43.18	18.13	88.69	85.14
FTP/LingtFTP	81.12	89.23	10.77	18.88	93.52	86.88
RTSP/Live555	82.94	88.03	11.97	17.06	96.48	89.20
SIP/Kamailio	83.57	72.00	28.00	16.43	95.79	89.26
SMTP/Exim	83.16	79.22	20.78	16.84	97.19	89.63
SSH/OpenSSH	82.23	75.00	25.00	17.77	98.16	89.49
TLS/OpenSSL	82.09	80.43	19.57	17.91	98.78	89.67
SSH/LibSSH	84.05	88.19	11.81	15.95	96.30	89.76

Conclusions

- We infer and discover the stateful variables used to store and reflect the states of network protocol software from the life cycle and operational characteristics of variables by analyzing prominent operations in the critical areas of memory from the process.
- We design and implement a prototype system for automatically discovering stateful variables in protocol software, and then perform experiments on nine programs in ProFuzzBench and two real-world programs. The average TPR can reach 82%, and the average precision can be up to about 96%.
- In the future, we may use symbolic taint analysis or conclic testing methods based on this knowledge, to assist the analysis tools in finding vulnerabilities more efficiently and accurately in network protocol software.

References

[1] Lee C, Bae J, Lee H, 2018. PRETT: protocol reverse engineering using binary tokens and network traces. Proc 33rd IFIP Int Conf on ICT Systems Security and Privacy Protection, p.141-155.

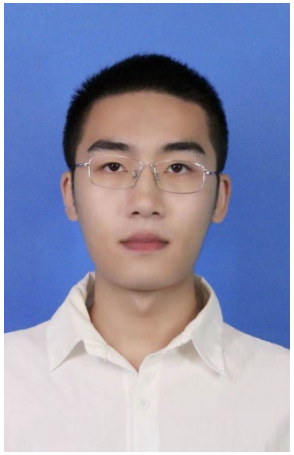
[2] Garmany B, Stoffel M, Gawlik R, et al., 2019. Static detection of uninitialized stack variables in binary code. Proc 24th European Symp on Research in Computer Security, p.68-87.

[3] Natella R, 2022. StateAFL: greybox fuzzing for stateful network servers. *Empir Softw Eng*, 27(7):191.

[4] Yu B, Wang PF, Yue T, et al., 2019. Poster: fuzzing IoT firmware via multi-stage message generation. Proc ACM SIGSAC Conf on Computer and Communications Security, p.2525-2527.

Jianxin HUANG received his BS and MS degrees from Information Engineering University, Zhengzhou, China, in network engineering and computer application technology, respectively. He is currently pursuing his PhD degree in cyberspace security from National University of Defense Technology, Changsha, China. His main research interests include network protocols and network security.

Bo YU received his BS degree from College of Computer, University of South China and his MS and PhD degrees from the College of Computer, National University of Defense Technology, Changsha, China, all in computer science, in 2007, 2010, and 2013 respectively. He is currently an associate professor with the College of Computer, National University of Defense Technology. His main research interests include software analysis, fuzzing technology, and IoT security.



Runhao LIU received his BS and MS degrees in computer science from the College of Computer, National University of Defense Technology, Changsha, China. He is currently a PhD candidate with the College of Computer, National University of Defense Technology. His research interests include software security, protocol security, and embedded security.

Jinshu SU received his BS degree in mathematics from Nankai University, Tianjin, China in 1985, and his MS and PhD degrees in computer science from National University of Defense Technology, Changsha, China in 1988 and 2000, respectively. He is a Professor in the College of Computer, National University of Defense Technology. His research interests include Internet architecture, network security, and AI for networking.